

FIDO Metadata Service

Proposed Standard, May 18, 2021



This version:

<http://fidoalliance.org/specs/mds/fido-metadata-service-v3.0-ps-20210518.html>

Issue Tracking:

[GitHub](#)

Editors:

[Billy Jack](#) (Microsoft)

[Rolf Lindemann](#) (Nok Nok Labs)

[Yuriy Ackermann](#) (FIDO Alliance)

Copyright © 2021 [FIDO Alliance](#). All Rights Reserved.

Abstract

The FIDO Authenticator Metadata Specification defines so-called "Authenticator Metadata" statements. The metadata statements contains the "Trust Anchor" required to validate the attestation object, and they also describe several other important characteristics of the authenticator. The metadata service described in this document defines a baseline method for relying parties to access the latest metadata statements.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current FIDO Alliance publications and the latest revision of this technical report can be found in the [FIDO Alliance specifications index](#) at <https://www.fidoalliance.org/specifications/>.

This document was published by the [FIDO Alliance](#) as a Proposed Standard. If you wish to make comments regarding this document, please [Contact Us](#). All comments are welcome.

Implementation of certain elements of this Specification may require licenses under third party intellectual property rights, including without limitation, patent rights. The FIDO Alliance, Inc. and its Members and any other contributors to the Specification are not, and shall not be held, responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.



THIS FIDO ALLIANCE SPECIFICATION IS PROVIDED “AS IS” AND WITHOUT ANY WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY EXPRESS OR IMPLIED WARRANTY OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

This document has been reviewed by FIDO Alliance Members and is endorsed as a Proposed Standard. It is a stable document and may be used as reference material or cited from another document. FIDO Alliance’s role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment.

Table of Contents

1	Notation
1.1	Key Words
2	Overview
2.1	Scope
2.2	Detailed Architecture
3	Metadata Service Details
3.1	Metadata BLOB Format
3.1.1	Metadata BLOB Payload Entry dictionary
3.1.2	BiometricStatusReport dictionary
3.1.3	StatusReport dictionary
3.1.4	AuthenticatorStatus enum
3.1.4.1	Certification Related Statuses
3.1.4.2	Security Notification Statuses
3.1.4.3	Info Statuses
3.1.5	RogueListEntry dictionary
3.1.6	Metadata BLOB Payload dictionary
3.1.7	Metadata BLOB
3.1.7.1	Examples
3.2	Metadata BLOB object processing rules
4	Considerations
	Index
	Terms defined by this specification
	Terms defined by reference

References

Normative References

Informative References

IDL Index

1. Notation§

Type names, attribute names and element names are written as code

String literals are enclosed in “”, e.g. “UAF-TLV”.

In formulas we use “|” to denote byte wise concatenation operations.

The notation `base64url(byte[8..64])` reads as 8-64 bytes of data encoded in base64url, "Base 64 Encoding with URL and Filename Safe Alphabet" [\[RFC4648\]](#) *without padding*.

Following [\[WebIDL-ED\]](#), dictionary members are optional unless they are explicitly marked as required.

WebIDL dictionary members MUST NOT have a value of null.

Unless otherwise specified, if a WebIDL dictionary member is DOMString, it MUST NOT be empty.

Unless otherwise specified, if a WebIDL dictionary member is a List, it MUST NOT be an empty list.

For definitions of terms, please refer to the FIDO Glossary [\[FIDOGlossary\]](#).

All diagrams, examples, notes in this specification are non-normative.

Note: Certain dictionary members need to be present in order to comply with FIDO requirements. Such members are marked in the WebIDL definitions found in this document, as required. The keyword `required` has been introduced by [\[WebIDL-ED\]](#), which is a work-in-progress. If you are using a WebIDL parser which implements [\[WebIDL\]](#), then you may remove the keyword `required` from your WebIDL and use other means to ensure those fields are present.

1.1. Key Words§

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [\[RFC2119\]](#).

2. Overview§

This section is not normative.

[\[FIDOMetadataStatement\]](#) defines authenticator metadata statements.

These metadata statements contain the trust anchor required to verify the attestation object (more specifically the KeyRegistrationData object), and they also describe several other important characteristics of the authenticator, including supported authentication and registration assertion schemes, and key protection flags.

These characteristics can be used when defining policies about which authenticators are acceptable for registration or authentication.

The metadata service described in this document defines a baseline method for relying parties to access the latest metadata statements.

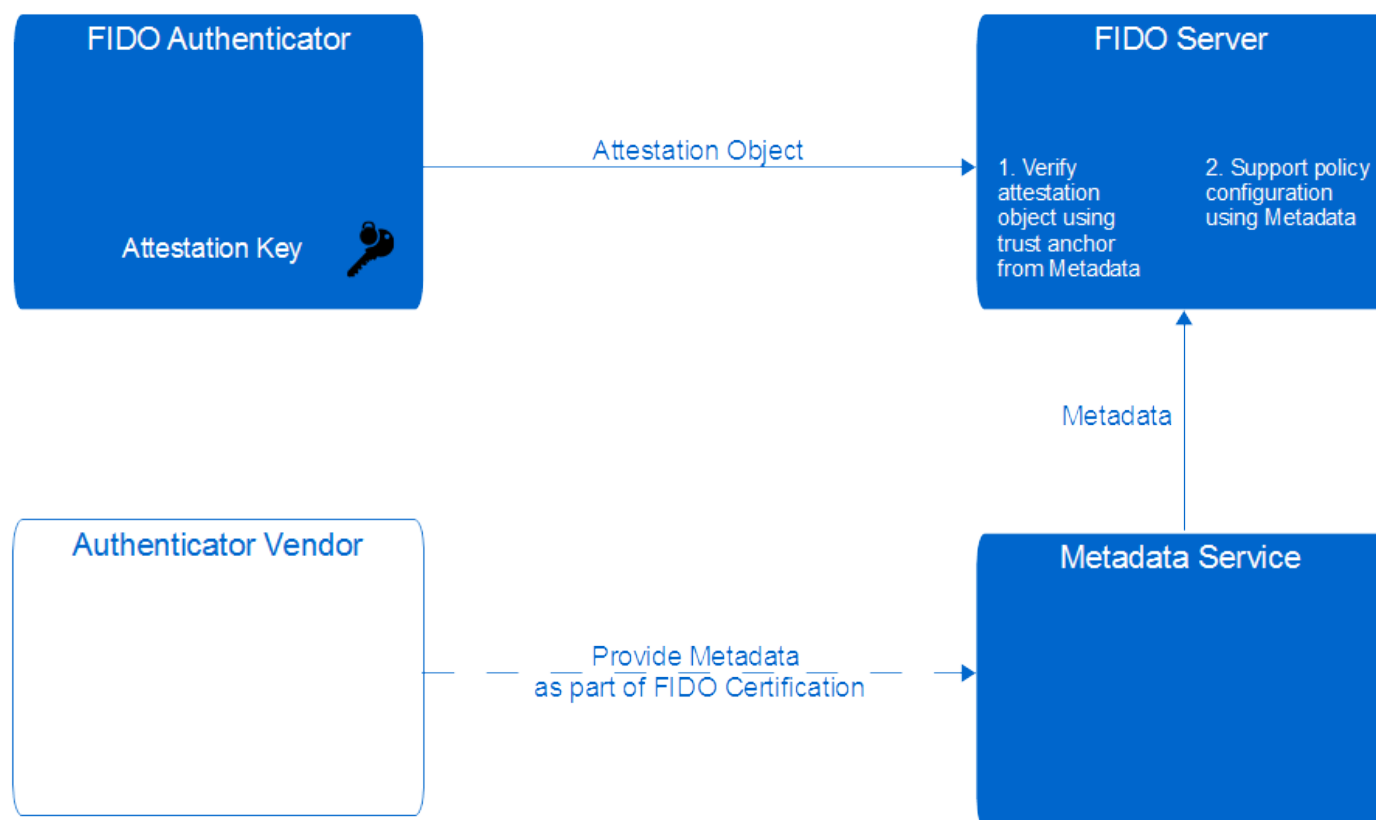


Figure 1 FIDO Metadata Service Architecture Overview

2.1. Scope§

This document describes the FIDO Metadata Service architecture in detail and it defines the structure and interface to access this service. It also defines the flow of the metadata related messages and presents the rationale behind the design choices.

2.2. Detailed Architecture§

The metadata BLOB file contains a list of metadata statements related to the authenticators known to the FIDO Alliance (FIDO Authenticators).

The FIDO Server downloads the metadata BLOB file from a well-known FIDO URL and caches it locally.

The FIDO Server verifies the integrity and authenticity of this metadata BLOB file using the digital signature. It then iterates through the individual entries and parses the metadata statements related to authenticator models relevant to the relying party.

Individual metadata statements are included in the entry of the metadata BLOB file, and may be cached by the FIDO Server as required.

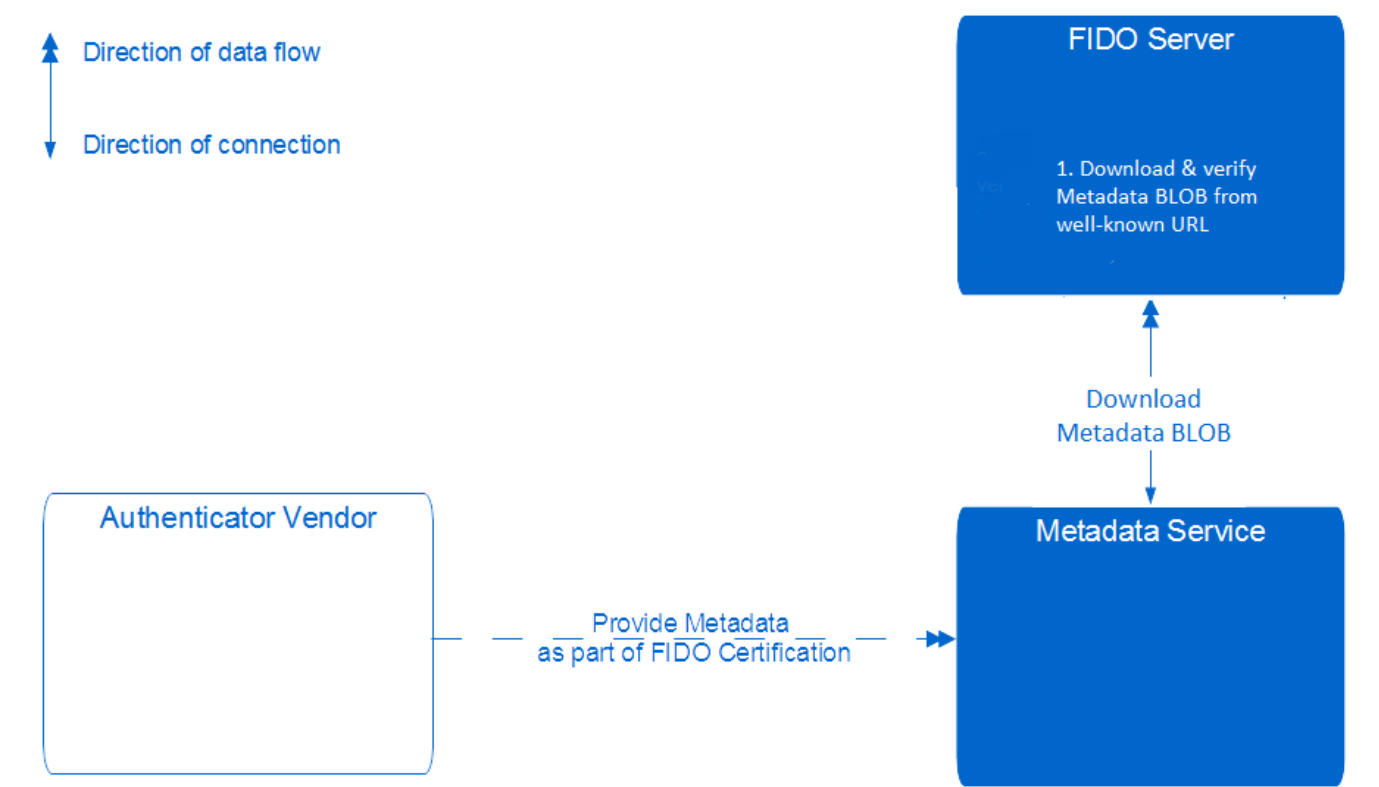


Figure 2 FIDO Metadata Service Architecture

The single arrow indicates the direction of the network connection, the double arrow indicates the

direction of the data flow.

The metadata BLOB file is accessible at a well-known URL published by the FIDO Alliance.

The relying party decides how frequently the metadata service is accessed to check for metadata BLOB updates.

3. Metadata Service Details§

This section is normative.

The relying party can decide whether it wants to use the metadata service and whether or not it wants to accept certain authenticators for registration or authentication.

The relying party could also obtain metadata directly from authenticator vendors or other trusted sources.

3.1. Metadata BLOB Format§

The metadata service makes the metadata BLOB object (see [Metadata BLOB](#)) accessible to FIDO Servers.

This object contains all metadata for each authenticator including the metadata statements defined in [FIDOMetadataStatement](#). The BLOB object contains one signature.

3.1.1. Metadata BLOB Payload Entry dictionary§

Represents the MetadataBLOBPayloadEntry

```
dictionary MetadataBLOBPayloadEntry {  
    AAID                aaid;  
    AAGUID              aaguid;  
    DOMString[]         attestationCertificateKeyIdentifiers;  
    MetadataStatement   metadataStatement;  
    BiometricStatusReport[] biometricStatusReports;  
    required StatusReport[] statusReports;  
    required DOMString   timeOfLastStatusChange;
```

```
DOMString      rogueListURL;  
DOMString      rogueListHash;  
};
```

aaid*, of type **AAID*

The AAID of the authenticator this metadata BLOB payload entry relates to. See [\[UAFProtocol\]](#) for the definition of the AAID structure. This field **MUST** be set if the authenticator implements FIDO UAF.

Note: FIDO UAF authenticators support AAID, but they don't support AAGUID.

aaguid*, of type **AAGUID*

The Authenticator Attestation GUID. See [\[FIDOKeyAttestation\]](#) for the definition of the AAGUID structure. This field **MUST** be set if the authenticator implements FIDO2.

Note: FIDO2 authenticators support AAGUID, but they don't support AAID.

attestationCertificateKeyIdentifiers*, of type **DOMString[]*

A list of the attestation certificate public key identifiers encoded as hex string. This value **MUST** be calculated according to method 1 for computing the keyIdentifier as defined in [\[RFC5280\]](#) section 4.2.1.2.

- The hex string **MUST NOT** contain any non-hex characters (e.g. spaces).
- All hex letters **MUST** be lower case.
- This field **MUST** be set if neither *aaid* nor *aaguid* are set. Setting this field implies that the attestation certificate(s) are dedicated to a single authenticator model.

FIDO U2F authenticators do not support AAID nor AAGUID, but they use attestation certificates dedicated to a single authenticator model.

metadataStatement*, of type **MetadataStatement*

The metadataStatement JSON object as defined in [\[FIDOMetadataStatement\]](#).

biometricStatusReports*, of type **BiometricStatusReport[]*

Status of the FIDO Biometric Certification of one or more biometric components of the Authenticator [\[FIDOBiometricsRequirements\]](#).

statusReports*, of type **StatusReport[]*

An array of status reports applicable to this authenticator.

timeOfLastStatusChange*, of type **DOMString*

ISO-8601 formatted date since when the status report array was set to the current value.

rogueListURL*, of type **DOMString*

URL of a list of rogue (i.e. untrusted) individual authenticators.

***rogueListHash*, of type [DOMString](#)**

`base64url(string[1..512])`

The hash value computed over the Base64url encoding of the UTF-8 representation of the JSON encoded `rogueList` available at `rogueListURL` (with type `rogueListEntry[]`). The hash algorithm related to the signature algorithm specified in the JWTHeader (see [Metadata BLOB](#)) MUST be used.

This hash value MUST be present and non-empty whenever `rogueListURL` is present.

This method of base64url-encoding the UTF-8 representation is also used by JWT [\[JWT\]](#) to avoid encoding ambiguities.

EXAMPLE 1

```
{
  "no": 1234,
  "nextUpdate": "2014-03-31",
  "entries": [
    {
      "aaid": "1234#5678",
      "metadataStatement": "Metadata Statement object as defined in Metadata Statement
spec.",
      "statusReports": [
        {
          "status": "FIDO_CERTIFIED",
          "effectiveDate": "2014-01-04"
        }
      ],
      "timeOfLastStatusChange": "2014-01-04"
    },
    {
      "attestationCertificateKeyIdentifiers": [
        "7c0903708b87115b0b422def3138c3c864e44573"
      ],
      "metadataStatement": "Metadata Statement object as defined in Metadata Statement
spec.",
      "statusReports": [
        {
          "status": "FIDO_CERTIFIED",
          "effectiveDate": "2014-01-07"
        },
        {
          "status": "UPDATE_AVAILABLE",
          "effectiveDate": "2014-02-19",

```

```

        "url": "https://example.com/update1234"
    }
],
"timeOfLastStatusChange": "2014-02-19"
}
]
}

```

3.1.2. BiometricStatusReport dictionary§

Contains the current BiometricStatusReport of one of the authenticator's biometric component.

```

dictionary BiometricStatusReport {
    required unsigned short certLevel;
    required DOMString      modality;
    DOMString               effectiveDate;
    DOMString               certificationDescriptor;
    DOMString               certificateNumber;
    DOMString               certificationPolicyVersion;
    DOMString               certificationRequirementsVersion;
};

```

certLevel, of type **unsigned short**

Achieved level of the biometric certification of this biometric component of the authenticator [\[FIDO Biometrics Requirements\]](#).

modality, of type **DOMString**

A *single* a single USER_VERIFY short form case-sensitive string name constant, representing biometric modality. See section "User Verification Methods" in [\[FIDO Registry\]](#) (e.g. "fingerprint_internal"). This value MUST NOT be empty and this value MUST correspond to one or more entries in field userVerificationDetails in the related Metadata Statement [\[FIDO Metadata Statement\]](#). This value MUST represent a biometric modality.

For example use USER_VERIFY_FINGERPRINT for the fingerprint based biometric component. In this case the related Metadata Statement must also claim fingerprint as one of the user verification methods.

effectiveDate, of type **DOMString**

ISO-8601 formatted date since when the certLevel achieved, if applicable. If no date is given, the status is assumed to be effective while present.

certificationDescriptor, of type **DOMString**

Describes the externally visible aspects of the Biometric Certification evaluation.

For example it could state that the "biometric component is implemented OnChip - keeping biometric data inside the chip only".

***certificateNumber*, of type [DOMString](#)**

The unique identifier for the issued Biometric Certification.

***certificationPolicyVersion*, of type [DOMString](#)**

The version of the Biometric Certification Policy the implementation is Certified to, e.g. "1.0.0".

***certificationRequirementsVersion*, of type [DOMString](#)**

The version of the Biometric Requirements [\[FIDO Biometrics Requirements\]](#) the implementation is certified to, e.g. "1.0.0".

3.1.3. StatusReport dictionary§

Contains an AuthenticatorStatus and additional data associated with it, if any.

New StatusReport entries will be added to report known issues present in firmware updates.

The latest StatusReport entry MUST reflect the "current" status. For example, if the latest entry has status USER_VERIFICATION_BYPASS, then it is recommended assuming an increased risk associated with all authenticators of this AAID; if the latest entry has status UPDATE_AVAILABLE, then the update is intended to address at least all previous issues *reported* in this StatusReport dictionary.

```
dictionary StatusReport {  
    required AuthenticatorStatus status;  
    DOMString effectiveDate;  
    unsigned long authenticatorVersion;  
    DOMString certificate;  
    DOMString url;  
    DOMString certificationDescriptor;  
    DOMString certificateNumber;  
    DOMString certificationPolicyVersion;  
    DOMString certificationRequirementsVersion;  
};
```

***status*, of type [AuthenticatorStatus](#)**

Status of the authenticator. Additional fields MAY be set depending on this value.

***effectiveDate*, of type [DOMString](#)**

ISO-8601 formatted date since when the status code was set, if applicable. If no date is given,

the status is assumed to be effective while present.

***authenticatorVersion*, of type [unsigned long](#)**

The authenticatorVersion that this status report relates to. In the case of FIDO_CERTIFIED* status values, the status applies to higher authenticatorVersions until there is a new statusReport.

For example, if the status would be USER_VERIFICATION_BYPASS, the authenticatorVersion indicates the vulnerable firmware version of the authenticator. Similarly, if the status would be UPDATE_AVAILABLE, the authenticatorVersion indicates the updated firmware version that is available now. If the status would be SELF_ASSERTION_SUBMITTED, the authenticatorVersion indicates the firmware version that the self assertion was based on.

***certificate*, of type [DOMString](#)**

Base64-encoded [\[RFC4648\]](#) (not base64url!) DER [\[ITU-X690-2008\]](#) PKIX certificate value related to the current status, if applicable.

As an example, this could be an Attestation Root Certificate (see [\[FIDOMetadataStatement\]](#)) related to a set of compromised authenticators (ATTESTATION_KEY_COMPROMISE).

***url*, of type [DOMString](#)**

HTTPS URL where additional information may be found related to the current status, if applicable.

For example a link to a web page describing an available firmware update in the case of status UPDATE_AVAILABLE, or a link to a description of an identified issue in the case of status USER_VERIFICATION_BYPASS.

***certificationDescriptor*, of type [DOMString](#)**

Describes the externally visible aspects of the Authenticator Certification evaluation.

For example it could state that the authenticator is a "SecurityKey based on a CC EAL 5 certified chip hardware".

***certificateNumber*, of type [DOMString](#)**

The unique identifier for the issued Certification.

***certificationPolicyVersion*, of type [DOMString](#)**

The version of the Authenticator Certification Policy the implementation is Certified to, e.g. "1.0.0".

***certificationRequirementsVersion*, of type [DOMString](#)**

The Document Version of the Authenticator Security Requirements (DV)

[\[FIDOAuthenticatorSecurityRequirements\]](#) the implementation is certified to, e.g. "1.2.0".

3.1.4. AuthenticatorStatus enum§

This enumeration describes the status of an authenticator model as identified by its AAID/AAGUID or attestationCertificateKeyIdentifiers and potentially some additional information (such as a specific attestation key).

```
enum AuthenticatorStatus {  
    "NOT_FIDO_CERTIFIED",  
    "FIDO_CERTIFIED",  
    "USER_VERIFICATION_BYPASS",  
    "ATTESTATION_KEY_COMPROMISE",  
    "USER_KEY_REMOTE_COMPROMISE",  
    "USER_KEY_PHYSICAL_COMPROMISE",  
    "UPDATE_AVAILABLE",  
    "REVOKED",  
    "SELF_ASSERTION_SUBMITTED",  
    "FIDO_CERTIFIED_L1",  
    "FIDO_CERTIFIED_L1plus",  
    "FIDO_CERTIFIED_L2",  
    "FIDO_CERTIFIED_L2plus",  
    "FIDO_CERTIFIED_L3",  
    "FIDO_CERTIFIED_L3plus"  
};
```

3.1.4.1. Certification Related Statuses§

NOT_FIDO_CERTIFIED

This authenticator is not FIDO certified.

Applicable StatusReport fields are:

- effectiveDate - When status was achieved
- authenticatorVersion - The minimum applicable authenticator version.
- url - To the authenticator page or additional information about the authenticator

SELF_ASSERTION_SUBMITTED

The authenticator vendor has completed and submitted the self-certification checklist to the FIDO Alliance. If this completed checklist is publicly available, the URL will be specified in url.

Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - New authenticator version that is

FIDO_CERTIFIED

This authenticator has passed FIDO functional certification. This certification scheme is phased out and will be replaced by FIDO_CERTIFIED_L1.

Applicable StatusReport fields are:

- effectiveDate - When certification was issued
- authenticatorVersion - The minimum version of the certified solution
- certificationDescriptor - Authenticator Description. I.e. "Munikey 7c Black Edition"
- certificateNumber - FIDO Alliance Certificate Number
- certificationPolicyVersion - Authenticator Certification Policy
- certificationRequirementsVersion - Security Requirements Version
- url - URL to the certificate, or the news article about achievement of the certification.

These fields are applicable to any of the FIDO_CERTIFIED_.*.

FIDO_CERTIFIED_L1

The authenticator has passed FIDO Authenticator certification at level 1. This level is the more strict successor of FIDO_CERTIFIED.

FIDO_CERTIFIED_L1plus

The authenticator has passed FIDO Authenticator certification at level 1+. This level is the more than level 1.

FIDO_CERTIFIED_L2

The authenticator has passed FIDO Authenticator certification at level 2. This level is more strict than level 1+.

FIDO_CERTIFIED_L2plus

The authenticator has passed FIDO Authenticator certification at level 2+. This level is more strict than level 2.

FIDO_CERTIFIED_L3

The authenticator has passed FIDO Authenticator certification at level 3. This level is more strict than level 2+.

FIDO_CERTIFIED_L3plus

The authenticator has passed FIDO Authenticator certification at level 3+. This level is more

strict than level 3.

REVOKED

The FIDO Alliance has determined that this authenticator should not be trusted for any reason. For example if it is known to be a fraudulent product or contain a deliberate backdoor. Relying parties SHOULD reject any future registration of this authenticator model.

Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - New authenticator version that is
- url - URL to the news/corporate article explaining the reason for revocation

3.1.4.2. Security Notification Statuses§

USER_VERIFICATION_BYPASS

Indicates that malware is able to bypass the user verification. This means that the authenticator could be used without the user's consent and potentially even without the user's knowledge.

Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - Minimum affected authenticator version
- url - URL to the news/corporate article explaining the incident

ATTESTATION_KEY_COMPROMISE

Indicates that an attestation key for this authenticator is known to be compromised. The relying party SHOULD check the certificate field and use it to identify the compromised authenticator batch. If the certificate field is not set, the relying party should reject all new registrations of the compromised authenticator. The Authenticator manufacturer should set the date to the date when compromise has occurred.

Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - Minimum affected authenticator version
- certificate - Base64 DER-encoded PKIX certificate identifying compromised attestation root. If missing, then assume all authenticators of this model are compromised.
- url - URL to the news/corporate article explaining the incident

USER_KEY_REMOTE_COMPROMISE

This authenticator has identified weaknesses that allow registered keys to be compromised and should not be trusted. This would include both, e.g. weak entropy that causes predictable keys to be generated or side channels that allow keys or signatures to be forged, guessed or extracted.

Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - Minimum affected authenticator version
- url - URL to the news/corporate article explaining the incident

USER_KEY_PHYSICAL_COMPROMISE

This authenticator has known weaknesses in its key protection mechanism(s) that allow user keys to be extracted by an adversary in physical possession of the device.

Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - Minimum affected authenticator version
- url - URL to the news/corporate article explaining the incident

3.1.4.3. Info Statuses§

UPDATE_AVAILABLE

A software or firmware update is available for the device. The Authenticator manufacturer should set the url to the URL where users can obtain an update and the date the update was published. When this status code is used, then the field authenticatorVersion in the authenticator Metadata Statement [\[FIDOMetadataStatement\]](#) MUST be updated, if the update fixes severe security issues, e.g. the ones reported by preceding StatusReport entries with status code USER_VERIFICATION_BYPASS, ATTESTATION_KEY_COMPROMISE, USER_KEY_REMOTE_COMPROMISE, USER_KEY_PHYSICAL_COMPROMISE, REVOKED. The Relying party MUST reject the Metadata Statement if the authenticatorVersion has not increased

Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported

- authenticatorVersion - New authenticator version that is available. MUST match authenticatorVersion in the metadata statement.
- url - URL to the page with the update info

Relying parties might want to inform users about available firmware updates.

More values might be added in the future. FIDO Servers MUST silently ignore all unknown AuthenticatorStatus values.

3.1.5. RogueListEntry dictionary§

Contains a list of individual authenticators known to be rogue.

New RogueListEntry entries will be added to report new individual authenticators known to be rogue.

Old RogueListEntry entries will be removed if the individual authenticator is known to not be rogue any longer.

Contains a list of individual authenticators known to be rogue.

New RogueListEntry entries will be added to report new individual authenticators known to be rogue.

Old RogueListEntry entries will be removed if the individual authenticator is known to not be rogue any longer.

```
dictionary RogueListEntry {
    required DOMString sk;
    required DOMString date;
};
```

sk, of type DOMString

Base64url encoding of the rogue authenticator's secret key (sk value, see [\[FIDOEcdaaAlgorithm\]](#), section ECDAAttestation).

In order to revoke an individual authenticator, its secret key (sk) must be known.

date, of type DOMString

ISO-8601 formatted date since when this entry is effective.

 **EXAMPLE: ROGUELISTENTRY[] EXAMPLE**

```
[
  { "sk": "M0-oaqbeJSSayzXaDUhh9LMKeT4Zio1bqn6W8kDaUfM",
    "date": "2016-06-07"},
  { "sk": "k96Npt4jJIq7NNNoNSGH0swp5PhU6jVuyf5jyYNtxrNQ",
    "date": "2016-06-09"},
]
```

3.1.6. Metadata BLOB Payload dictionary§

Represents the MetadataBLOBPayload

```
dictionary MetadataBLOBPayload {
  DOMString legalHeader;
  required Number no;
  required DOMString nextUpdate;
  required MetadataBLOBPayloadEntry[] entries;
};
```

***LegalHeader*, of type DOMString**

The legalHeader, which MUST be in each BLOB, is an indication of the acceptance of the relevant legal agreement for using the MDS. The FIDO Alliance's Blob will contain this legal header: "legalHeader": "Retrieval and use of this BLOB indicates acceptance of the appropriate agreement located at <https://fidoalliance.org/metadata/metadata-legal-terms/>"

***no*, of type Number**

The serial number of this UAF Metadata BLOB Payload. Serial numbers MUST be consecutive and strictly monotonic, i.e. the successor BLOB will have a no value exactly incremented by one.

***nextUpdate*, of type DOMString**

ISO-8601 formatted date when the next update will be provided at latest.

***entries*, of type MetadataBLOBPayloadEntry[]**

List of zero or more MetadataBLOBPayloadEntry objects.

3.1.7. Metadata BLOB§

The metadata BLOB is a JSON Web Token (see [\[JWT\]](#) and [\[JWS\]](#)). It consists of three elements:

- The base64url encoding, without padding, of the UTF-8 encoded JWT Header (see example below),
- the base64url encoding, without padding, of the UTF-8 encoded Metadata BLOB Payload (see example at the beginning of section [Metadata BLOB Format](#)),

- and the base64url-encoded, also without padding, JWS Signature [\[JWS\]](#) computed over the to-be-signed payload using the Metadata BLOB signing key, i.e. `tbSPayload = EncodedJWTHeader | "." | EncodedMetadataBLOBPayload`

All three elements of the BLOB are concatenated by a period ("."): `MetadataBLOB = EncodedJWTHeader | "." | EncodedMetadataBLOBPayload | "." | EncodedJWSSignature`

The hash algorithm related to the signing algorithm specified in the JWT Header (e.g. SHA256 in the case of "ES256") MUST also be used to compute the hash of the metadata statements (see section [Metadata BLOB Payload Entry Dictionary](#)).

3.1.7.1. Examples[§]

This section is not normative.

EXAMPLE: ENCODED METADATA BLOB

```
ewoJImx1Z2FsSGVhZGVyIjogIlJldHJpZXZhbCBhbmQgdXNlIG9mIHRoaXMgQkxPQiBpbmRpY2F0ZXMg
YWNjZXB0YW5jZSBvZiB0aGUgYXBwcm9wcm1hdGUgYWdyZWVtZW50IGxvY2F0ZWQgYXQgaHR0cHM6Ly9m
aWRvYXNsaWZlY2Uub3JnL21ldGFkYXRhL21ldGFkYXRhLWx1Z2FsLXRlcm1zLyIsCgkibm8iOiAxNSwK
CSJuZXh0VXBkYXRlIjogIjIwMjAtMDMtMzAiLAoJImVudHJpZXMiOiBbewoJCQkiYWpZCI6ICIXMjM0
IzU2NzgiLAoJCQkibWV0YWRhdGF0ZGF0Zl1bnQiOiB7CgkJCQkibGVnYXZIZWFkZXIiOiAiaHR0cHM6
Ly9maWRvYXNsaWZlY2Uub3JnL21ldGFkYXRhL21ldGFkYXRhLXN0YXRlbWVudC1sZWdhbC1oZWfkZXIv
IiwKCQkJCJSjKjXNjcmlwdGlvbiI6ICJGSURPIEFsbGhbmNlIFNhbXBsZSBVQUYgQXV0aGVudG1jYXRv
ciIsCgkJCQkiYWpZCI6ICIXMjM0IzU2NzgiLAoJCQkJImFsdGVybmF0aXZlRGVzY3JpcHRpb25zIjog
ewoJCQkJCJSjYdS1SVSI6ICLQn9GA0LjQvNC10YAgVUFGINCw0YPRgtC10L3RgtC40YTQuNC60LDRgtC-
0YDQsCDQvtGCIeZJRE8gQWxsawFuY2UiLAoJCQkJCJSjmc1GUiI6ICJFeGVtcGx1IFVBRiBhdXR0ZW50
aWNhdG9yIGRlIEZJRE8gQWxsawFuY2UiCgkJCQk19LAoJCQkJImF1dGh1bnRpY2F0b3JWZXJzaW9uIjog
MiwKCQkJCJSjwcm90b2NvbEZhbWlseSI6ICJ1YWYiLAoJCQkJInNjaGVtYSI6IDMsCgkJCQkidXB2Ijog
W3sKCQkJCQkJIm1ham9yIjogMSwKCQkJCQkJIm1pbm9yIjogMAoJCQkJCX0sCgkJCQkJeWwJCQkJCQki
bWV0b3IiOiAxLAoJCQkJCQkibWlub3IiOiAxCgkJCQkJfQoJCQkJXSwwKCQkJCJSjhdXR0ZW50aWNhdGlv
bkFsZ29yaXRobXMiOiBbInNlY3AyNTZyMV9lY2RzYV9zaGEyNTZfcml0sCgkJCQkicHVibGljS2V5
QWxnQW5kRW5jb2RpbmdzIjogWyJlY2NfeDk2Ml9yYXciXSwwKCQkJCJSjhdHRlc3RhdGlvb1R5cGVzIjog
WyJiYXNpY19mdWxsIl0sCgkJCQkidXNlc1Zlcm1maWNhdGlvbkRldGFpbHMiOiBbCgkJCQkJeWw3sKCQkJ
CQkJInVzZXJWZXJpZmljYXRpb25NZXRob2QiOiAiZmluZ2VycHJpbnRfaw50ZXJuYWwiLAoJCQkJCQki
YmFEZXNjIjogewoJCQkJCQkJInNlbGZBdHRlc3RlZEZBUiI6IDAuMDAwMDIsCgkJCQkJCQkibWF4UmV0
cmllcyI6IDUsCgkJCQkJCQkiYmxvY2tTbG93ZG93biI6IDMwLAoJCQkJCQkJIm1heFRlbXBsYXRlcYI6
IDUKCQkJCQkJfQoJCQkJCX1dCgkJCQldLAoJCQkJImtleVByb3RlY3Rpb24iOiBbImhhcmR3YXJlIiwg
InRlZSjdLAoJCQkJIm1sZ2V5UmVzdHJpY3RlZCI6IHRydwUsCgkJCQkibWF0Y2h1clByb3RlY3Rpb24i
OiBbInRlZSjdLAoJCQkJImNyeXB0b1N0cmVuZ3RoIjogMTI4LAoJCQkJImF0dGFjaG11bnRlIaw50Ijog
WyJpbnRlcm5hbCJdLAoJCQkJInRjRGlzcGxheSI6IFsiYW55IiwgInRlZSjdLAoJCQkJInRjRGlzcGxh
eUNvbRlbnRUeXB1IjogIm1tYWdlL3BuZyIsCgkJCQkidGNEaXNwbGF5UE5HQ2hhcmFjdGVyaXN0aWNz
IjogW3sKCQkJCQkid2lkdggiOiAzMjAsCgkJCQkJImhlaWdodCI6IDQ4MCwKCQkJCQkiYml0RGVwdGgi
OiAxNiwwKCQkJCQkiY29sb3J1eXB1IjogMiwKCQkJCQkiY29tchJlc3Npb24iOiAwLAoJCQkJCJSjmaWx0
ZXIiOiAwLAoJCQkJCJSjpbmRlcmxhY2UiOiAwCgkJCQk19XSwwKCQkJCJSjhdHRlc3RhdGlvb1Jvb3RDZXJ0
```

aWZpY2F0ZXMiOiBbCgkJCQkJKik1JSUNQVENDQWVPZ0F3SUJBZ0lKQU91ZXh2VTNPeTJ3TUFvR0NDcUdT
TTQ5QkFNQ01Ic3hJREFlQmdOVkJBTU1GMU5oY1hCc1pTQkJKSFJsYzNSaGRHbHZiaUJTYjI5ME1SWXdG
QVlEVlFRS0RBMUdTVVJQSUVGc2JHbGhibU5sTVJFd0R3WURWUVMREFoVlFVWdWRmRITERFU01CQUdB
MVVFQnd3S1VHRnNieUJCykSdK1Rc3dDUVlEVlFRSURBSkRRVEVMTUFR0ExVUVCaE1DVlZNd0hoY05N
VFF3TmPFNE1UTXpNek15V2hjTk5ERXhNVEF6TVRNek16TXlXakI3TVNBd0hnWURWUWFEREJkVFlMXdi
R1VnUVhSMFpYtjBZWfJwYjI0Z1VtOXZkREVXTUJRR0ExVUVDZ3dOUmtsRVR5QkjiR3hwwVc1alpURVJN
QThHQTFVRUN3d0lWVUZHSUZWfJ5d3hFakFRQmdOVkKJBY01DVkJoYk4Z1FXeDBiekVMTUFR0ExVUVD
QXddUTBFeEN6QUpCZ05WQkFZVEFsVlRNRmt3RXdZSEtVWk16ajBDQVFZSUtVWk16ajBEQVFjRFFnQUVI
OGh2MkQwSFhNTkvQm1wUTdSWmVoTC9GTUd6RmQxUUJnOXZBVXBPWjNham51UTk0UFI3YU16SDMzb1VT
QnI4ZkhZRHJxT0JiNThweEdxSEpSeVgVnK5RTU00d0hRWURWUjBPQkJZRUZQb0hBM0NMaHhGyKmwSXQ3
ekU0dzhoazVFSi9NQjhHQTFVZE13UVlNQmFBR1BvSEezQ0xoeEZiQzBjdDd6RTR30GhrNUVKL01Bd0dB
MVVKRXdRRk1BTUJBJzh3Q2dZSUtVWk16ajBFQXDJRfNBQXdsU0UloQUowNlFTWHQ5aWhJYkVLWUtJanNQ
a3JpVmRMSWd0ZnNiRfN1N0VySmZ6cJRbaUJxb1lDwmYwK3pJNTVhUWVBSGpJekE5WG02M3JydUF4Qlo5
cHM5ejJYTMxRPT0iCgkJCQldLAoJCQkJKIm1jb24iOiAiZGF0YTppbWFnZS9wbmc7YmFzZTY0LG1WQk9S
dzBLR2dvQUFBQU5TVWhFVWdBQUFFOEfBQUF2Q0FZQUFBQU213SmZjQUFBQUFYt1NSME1BcnM0YzZRQUFB
QVJuUUVUxQkFBQ3hq3Y4WVfVQUFBQUjRWhaY3dBQURzTUFBQTFdEQWNkdFhUUFBUWFOU1VSQlZHaEQ3
WnI1YnhSbEdNZj1lLe1RCOEfNL1lFaEUyVzdWUVpjV0tLQmNsU3BIQVRsRUxBUKU3a05FQ0NBM0ZrV0sw
Q0tLU0NGSXNLQmNnVKNV0dORVnKQVlpZHdnZ2dKQm1SaU1oRmMvNHd5ODg4NHp1OU5kbG5HVGAzS1Ay
bjNuTysrODg5MzNmmdVCQngrUHFDeKprVFV2QmJMBXBVRFd2Q1RjBjBjQ1Nad1hMQ2RYOVIwNVNRMtli
YjVhdGY10TlMRysvZXJBNTQxcTQ3YVAXTEwYt1TSXlWt1VpOE1pOGQ1a0dUc2kzME5GdjhaTluN1Fa
UE13YmR5czJlclUyWE1xVWR5OCtaY2F0bUdpbU04eVhOM1JVZDNhMThuRjBmVWxvd1orMENUeldwZDJW
ait1T20xYkV5eTZEEDRpNXBVTUdXdmVvNTA2cTIyN2R0dVdCSXVmZnI2b1dwVjBGUE5MaG93MTc1MU5t
MjFmd1BIM3JWdFdqZno2NkxmcWw4dFg3R1Js0V1GU1hzbVnZWI5Y2VPR2JZazdNT1VjR1Bn0FpzYk1l
OXJmUVVhYVYvSk1Y0XNzZHpEQ1N2cDBrWkhtVFpnOXg3YkxIY01uVGhiMTZ1SittVmZRcTh5YVVAUU5H
NjRpwForMC9rcTZ1T1pGTzBRdGF0ZFdlZlhuU1E5OUJqOTFSNU9JRm5rNTRqTjBta1VpcWxPM1hEVytN
bCs5OG1LQjZ0VzdyV3BaY1BjKzB6ZzR0THJZbFVjODZFNmVHRGpJTXViVnBjdXNlYXJmZ0lZR1JrNmJy
aFpWci9KY0h6b29MNzU1MGplZEExFeG9wV2NBcGkyWlVxaHU3Skx2clZzUVU4MXprek9QZWVtTVJZdlZ1
UXNYN1BiaURRWTVKdlpvpbmZ0SysxVlk4SD1lDHg1MzBoMG9iK2ptU1lxajZvdWFZdkVlb1cvV2xZanA4
Y3diTW02ODJ0UHdxVzFSNHRqLzJTSDEzSVJKWw0bW9ad1hwaVNxRHI3ZFh0UUh4YS9QSzMvK0JXc0sx
ZFRnSHU2Vjh0U0ozYndGa3dwRnJVT1E1MHMxcjNsZXZtOHpaY3ExNytCQmF3N0s4bEVLNXF6a11lYXJr
OUE4cDdQM0d6REsrbmQzRFFvdys2VUM4U1Z0ODJpdXYzOG1tN050YVh0VjFDVnE2Umd3NHBrc21iZGkz
YnUyRGU3WwZhQk1Y4Y3FmdnFQclVqR1FOVFEyMmxmZfVWVlQ2OHJUSktGNURuU21VamdKcWc0bVNT0XBt
c2ZES1IzRzZub0gwaVc5YVY3TFdMSF1YS2xsVER0MEExUQXRrWUlhYw1wMVfQVnYrK3V5R1V4VmRKMERO
VlhTbStiMXFSeHBsODRkZGZYMuxwMU8vZDY5dHNvZDB2czVoR3JlOXh1OG8rZnBMUjFjR2hOVEQ2WjU3
Qz1LTvdYZWZKZE9aOTRiYj1vcWQxUk9uUzdXSVRUekhpBU1xaXZiTzNnMERKvNlrM1drQmhCenRLMzVZ
S05kt25j0E8zYWNTNmZEWkZnS2FYTHNFSnA1cmRybGlCcXA40WNKY3MvbTdUdnMwcmqtqR2ZONGIwa1Bv
Wm4zVUp1SU9ybloyMnlQMwZtd1V4K081Z1NxZWJWMM0re1N1WU5WaHE3VFdiRG1MVnZsanBsTGxvcDZD
TFhQKzJxdHZHTE1MLzF2aW1JU2RNQmd6U29Gwn11NlRxZCtqenhnc1BhVjlCQ3F1ZS90a11rNnY2bEs5
Y3dpVWmvU1R0ZjFIRHBNM2I10TJ5N2gzVGh4NW96SszY5SExwWVd1QXdhcVM1Y3YyNnE3Y2Vi0GVmV1lh
UmVQM21GVTh6ajFrb1N3WlhITw1uQ2pZME9nYwXvN1VRZ1NDTTNxFyMkgvWEZQN3NzWHg0NVlsOTFC
eWVDZXA0bW9ab0grMWZHM3hENHRUN3g4a3d5ajhud2I5ZXyYn1YwQjZkKzdINHPldnVkJUG1MzdGanF5
ek9IZEpuSEV1em1YcS9XanhPYnZOTWJ2N25oeXdxWDJhVnNXdEM4KzQ4YUx1YXBfN3A1d0taaTBBMkFR
U1Y1bnZSNEUrdUpjK2I2MwtBcHFJbnhCZ21kLzRWNVFQL210MThIREM3c1JIZnRtZXU1bG1oVjBybi9B
TFgyMzJicWQ0QkZuRHg3VmkxY1dTmNVMzjBJYkI0N3FlEHhtVwo5UXV0WwP1cGQzdFlENmfiV0JCTXJo
K2FwTmJPS3JORjErdWdDYTRYaVhHZndNUFB0Vm1hdmhVM1lNT0FBbnVvYi9SMDdMMH1PU2VPYWRFOdhB
cHNYRkdmZjMwew5obEpnTTUxQ1U2dk45RXpnbB2SEJGVXlpVnJhZVBpd0o1M0RGNVpUWm5vbUV0Zzg1

a05VZDJvSmkyV3ByNE9tbWtmTjR4NHpIZmLWRmM4RHY4Tnp1aE5xT2lkaWxHdke2REd1ZVp3Tzc4QUFR
bjZjaUVrNitydzVWY3ZqdnFORFlQT29JVXdhS1NocnhBdVhMbGtINGFZdUdmTVlEYzEwV0Y1VGEzMWWhQ
Sk9mY1Voc1UvSmxJTMk2YzZlbfJZZEJwbzYrK1lmang2MWxHTmZSbTRNRDVySjFqM0ZvR0huakRTQk5h
c1lVZ01MeU1zektwYjd0WHBvSGZQczhoM1dwMux6TmZOazU0WHhDMXdER1VtWxpYwWVmaDZ6L2NLdFZt
NEVCeGE5V1FHRHpZcjNmclVNUmpIRUtrazd6YUZLWVFBMmhHUVUxeis4NU5GV3BYRHJrejN2eDEwR3F4
UTZCemVOYm9CazVuOGs0bmViUmgrazFoV2Z4VEYwRDFFeVdVczVuditkZ1FxS2F4enVDZEUwaXNIbDAY
TlE4YWgwbVhyMTJMYTNTMGY5d2lr0St3TE5UTVkvODZNUG84ewkzMU9meG1UNlBXb3FH0StEwnVrWW5h
NTZtU1p0NVdXU3k1cVZBMXJ3VXlKcVhBbG56a2lhaS9nSFNEN1JrVHlpaG9nQUFBQUJKULU1RXJrSmdn
Zz09IgoJCQl9LAoJCQkic3RhdHVzUmVwb3J0cyI6IFt7CgkJCQkic3RhdHVzIjogIkZJRE9fQ0VSVElG
SUVEIiwKCQkJCSJlZmZlY3RpdMVEYXRlIjogIjIwMTQtMDEtMDQiCgkJCX1dLAoJCQkidGltZU9mTGFz
dFN0YXRlc0NoYW5nZSI6ICImDE0LTaxLTA0IgoJCX0sCgkJewoJCQkiYwFndWlkIjogIjAxMzJkMTEw
LWJmNGUtdNDiW0C1hNDazLWFInGY1ZjEyZWZlNSIsCgkJCSJtZXRhZGF0YVN0YXRlbWVudCI6IHsKCQkJ
CSJsZWdhbEh1YWRLciI6ICJodHRwczovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV0YWRhdGEt
c3RhdGVtZW50LWxlZ2F5LWwh1YWRLci8iLAoJCQkJImRlc2NyaXB0aw9uIjogIkZJRE8gQWxsawFuY2Ug
U2FtcGx1IEZJRE8yIEF1dGhlbnRpy2F0b3IiLAoJCQkJImFhZ3VpZCI6ICImMTMyZDExMzIjZjRlLTQy
MDgtYTQwMy1hYjRmNWYxMmVmZTUilAoJCQkJImFsdGVybmF0aXZlRGVzY3JpcHRpb25zIjogewoJCQkJ
CSJydS1SVSI6ICLQn9GA0LjQvNC10YAgRklETzIg0LDRg9GC0LXQvdGC0LjRhNC40LrQsNGC0L7RgNCw
INC-0YIgrKlETyBBbGxpYW5jZSI6CgkJCQkJImZyLUZSIjogIkV4ZW1wbGUgRklETzIgYXV0aGVudG1j
YXRvc1BkZSBGSURPIEFsbG1hbmNlIiwKCQkJCQkiemgtQ04iOiAi5L6G6IeqRklETyBBbGxpYW5jZee
hOekuuS-i0ZJRE8y6Lqr5Lu96amX6K2J5ZmoIgoJCQkJfSwKCQkJCSJwcm90b2NvbEZhbWlseSI6ICJm
aWRvMiIsCgkJCQkic2NoZW1hIjogMywKCQkJCSJhdXRoZW50aWNhdG9yVmVyc2lubiI6IDUsCgkJCQki
dXB2IjogW3sKCQkJCQkibWfqb3IiOiAxLAoJCQkJCSJtaW5vciI6IDAKCQkJCX1dLAoJCQkJImF1dGhl
bnRpy2F0aw9uQWxnb3JpdGhtcyI6IFsic2VjcDI1NnIxX2VjZHNhX3NoYTI1Nl9yYXciLCAicnNhc3Nh
X3BrY3N2MTVfc2hhMjU2X3JhdyJdLAoJCQkJInB1YmXpY0tleUFsZ0FuZEVuY29kaw5ncyI6IFsiY29z
ZSJdLAoJCQkJImF0dGVzdGF0aw9uVHlwZXMiOiBbImJhc2ljX2Z1bGwiXSwwKCQkJCSJlc2VyVmVyaWZp
Y2F0aw9uRGV0YWlscyI6IFsKCQkJCQlbewoJCQkJCQkidXNlc1Zlcm1maWNhdGlvbk1ldGhvZCI6ICJu
b25lIgoJCQkJCX1dLAoJCQkJCVt7CgkJCQkJCSJlc2VyVmVyaWZpY2F0aw9uTWV0aG9kIjogInByZXNl
bmNlX2ludGVybmFsIgoJCQkJCX1dLAoJCQkJCVt7CgkJCQkJCSJlc2VyVmVyaWZpY2F0aw9uTWV0aG9k
IjogInBhc3Njb2RlX2V4dGVybmFsIiwKCQkJCQkJImNhRGVzYyI6IHsKCQkJCQkJCSJiYXNlIjogMTAs
CgkJCQkJCQkibWlUTGVuZ3RoIjogNAoJCQkJCQl9CgkJCQkJfV0sCgkJCQkJW3sKCQkJCQkJCSJlc2Vy
VmVyaWZpY2F0aw9uTWV0aG9kIjogInBhc3Njb2RlX2V4dGVybmFsIiwKCQkJCQkJCSJjYURlc2MiOiB7
CgkJCQkJCQkJImJhc2UiOiAxMwKCQkJCQkJCQkibWlUTGVuZ3RoIjogNAoJCQkJCQkJfQoJCQkJCQl9
LAoJCQkJCQl7CgkJCQkJCQkidXNlc1Zlcm1maWNhdGlvbk1ldGhvZCI6ICJwcmVzZW5jZV9pbmRlcm5h
bCIKCQkJCQkJfQoJCQkJCV0KCQkJCV0sCgkJCQkia2V5UHJvdGVjdGlvbiI6IFsiaGFyZDhcmUiLCAi
c2VjdXJlX2VsZW1lbnQiXSwKCQkJCSJtYXRjaGVyUHJvdGVjdGlvbiI6IFsiaGFyZDhcmUiLCAi
ImNyeXB0b1N0cmVuZ3RoIjogMTI4LAoJCQkJImF0dGFjaG1lbnRIaw50IjogWyJleHRlc2h0bCI6ICJ3
aXJlZCI6ICJ3aXJlbGVzcyIsICJmZmMlXSwKCQkJCSJ0Y0Rpc3BsYXkiOiBbXSwKCQkJCSJhdHRlc3Rh
dGlvb1Jvb3RDZXJ0awZpY2F0ZXMiOiBbCgkJCQkJIk1JSUNQVENDQWVPZ0F3SUJBZ0lKQU91ZXh2VTNP
eTJ3TUFvR0NDcUdTTTQ5QkFNQ001Ic3hJREFlQmdOVkKJBTU1GMU5oY1hCc1pTQkJkSFJsYzNSaGRHbHZi
aUJTYjI5ME1SWXdGQVlEVlFRS0RBMUdTVVJQSUVGc2JHbGhibU5sTVJFd0R3WURWUVFMRERoVlFVWdW
RmRITERFU01CQUdBMVVFQnd3S1VHRnNieUJCYkhSdK1Rc3dDUVlEVlFRSURBSKRREVMtUFR0EXVUVC
aE1DVlZNd0hoY05NVFF3TmPfNE1UTXpNek15V2hjTk5ERXhNVEF6TVRNek16TXlXakI3TVNBd0hnbWURW
UVFEREJkVFlMXdiR1VnUVhSMFpYTjBZWfJwYjI0Z1VtOXZkREVXTUJRR0ExVUVDZ3d0UmtsRVR5QkJi
R3hwwVc1alpURVJNQThHQTFRUN3d0lWVUZHSUZSWFJ5d3hfakFRQmdOVkKJBY01DVkJoYkZ4Z1FXeDBi
ekVMTUFR0EXVUVDQXdDUTBFeEN6QUpcZ05WQkFZVEFsVlRNRmt3RXdZSEtVWk16ajBDQVFZSUtVWk16
ajBEQVFfjRFFnQUVIOGh2MkQwSFhhNTkvQm1wUTdSwVoTC9GTUd6RmQxUUJnOXZBVXBPWjNham51UTk0

UFi3YU16SDMzb1VTQnI4ZkhZRHJxT0JiNThweEdxSEpSeVgvNk5RTUU0d0hRWURWUjBPQkJZRZUQb0hB
M0NMaHhGYkMwSXQ3ekU0dzhoazVFSi9NQjhHQTfVZE13UVlNQmFBR1BvSEeZQ0xoeEZiQzBJdDd6RTR3
OGhrNUVKL01Bd0dBMVVKRXdRRk1BTUJBZjh3Q2dZSutvWk16ajBFQXdJRFNBQXdSUUloQUowN1FTWHQ5
aWhJYkVLWUtJanNQa3JpVmRMSWd0ZnNiRFN1N0VySmZ6cjRBaUJxb1lDwmYwK3pJNTVhUWVBSGpJekE5
WG02M3JydUF4Q1o5cHM5ejJYTmxRPT0iCgkJCQlDLaOJCQkJIm1jb24iOiAiZGF0YTppbWFnZS9wbmc7
YmFzZTY0LG1WQk9SdzBLR2dvQUFBQU5TVWhFVWdBQUFFOEfBQUF2Q0FZQUFBQ213SmZjQUFBQUFYt1NS
ME1BcnM0YzZRQUFBQVJuUVUxQkFBQ3hq3Y4WfVQUFBQUFpJRWhaY3dBQURzTUFBQTdEQWNkdNFHUUFB
QWFoU1VSQlZHaEQ3WnI1YnhSbEdNZjlLe1RCOEfNL1lFaEUyVzdWUVpjV0tLQmNsU3BIQVRsRUxBUKU3
a05FQ0NBm0ZrV0swQ0tLU0NGSXNLQmNnVkNEV0d0RVNkQVlpZHdnZ2dKQm1SaU1oRmMvNhd50Dg4NHp1
OU5kbG5HVGAZs1A1ybJNuTysrODg5MzNmDMVCQngrUHFDeKprVFV2QmJMbXBVRfd2Q1RJBxBjQ1Nad1hM
Q2RYOVIwNVNrMTliYjVhdGY1OTlmRysvZXJBNTQxcTQ3YVAxTExWYT1TSXlWTlVp0ElpOGQ1a0dUc2kz
ME5GdjdhATlun1FaUE13YmR5czJlclUyWE1xVWR50CtaY2F0bUdpbUU4eVhOM1JVZDNhMTThuRjBmVwxv
dlorMEnUe1dwZDJWaiTlT20xYkV5eTZEeDRpNXBVTUdXdmVvNTA2cTIyN2R0dVdCSXVmZnI2b1dwVjBG
UE5MaG93MTc1MU5tMjFMdlBIM3JWdFdqZno2NkxmcWw4dFg3RlJsOv1GU1hzbVNzZWl5Y2VPR2JZazdN
TlVjR1Bn0FpzYk1lOXJmUVVhYVYvSk1YOXNxZHPeQ1N2cDBrWkhtVFpnOXg3YkxIY01uVghImTZlSitt
VmZRCTh5YVVAU5HNjRpWForMC9rcTZ1T1pGTzBRdGF0ZFdLZlhuU1E5OUJq0TFSNU9JRm5rNTRqTjBt
a1VpcWxPM1hEVytNbCs5OG1LQjZ0VzdyV3BaY1BjKzB6ZzR0THJZbFVjODZFNmVHRGpJTXViVnBjdXN1
YXJmZ0lZR1JrNmJyaFpWci9KY0h6b29MNzU1MGplZEExFeG9wV2NBcGkyWlVxaHU3Skx2c1ZzUVU4MXpr
ek9QZwVtTVJZd1Z1UXNYN1BiaURRWTVKdlpvbmZ0SysxV1k4SD11dHg1MzBoMG9iK2ptU1lXajZvdWFZ
dkV1blcvV2xZanA4Y3diTW02ODJ0UHdxVzFSNHRqLzJTSDEzSVJKWww0bW9ad1hwaVNxRHI3ZFh0UUh4
YS9QSzMvK0JXc0sxZFRnSHU2Vjh0UUozYndGa3dwRnJVT1E1MHMxcjNsZXZt0HpaY3ExNytCQmF3N0s4
bEVLNXF6a11lYXJR0UE4cDdQM0d6REsrbmQzRFFvdys2VUM4U1ZOODJpdXYzOG1tN050YVh0VjFDVnE2
Umd3NHBrc21iZGkzYnUyRGU3WWZhQkY4Y3FmdnFQc1VqRlFOVFEyMmxmZfVWVlQ20HJUSktGNURuU21V
amdKcWc0bVNT0XBtc2ZESlIzRzZUb0gwaVc5YVY3TFdMSf1YS2xsVER0MExUQXRrWUlhyW1wMVfQvNyr
K3V5R1V4VmRKMEROV1hTbStiMXFSeHBSODRkZGZYMUXwMU8vZDY5dHNvZDB2czVoR3JlOXh10G8rZnBM
UjFjR2h0VEQ2WjU3Qz1LTvdYZWZKZE9aOTRiYj1vcWQxUk9uUzdXSVRUekhpBU1xaXZiTzNnMERkVn1r
M1dRQmhCenRLmzVZS05kT25jOE8zYWNtNmZEWkZnS2FYTHNfSnA1cmRybGlCcXA40WNKY3MvbTdUdnMw
cmtqR2ZONGIwa1BvWm4zVUp1SU9ybloyMn1QMwZtd1V4K081Z1NxZWJWMW0re1N1WU5WaHE3VFdiRG1M
VnZsanBsTGxvcDZDTfHqKzJxdHZHTE1MLzF2aw1JU2RNQmd6U29Gwn11N1RxZCtqenhnc1BhVjlCQ3F1
ZS90a1lrNnY2bEs5Y3dpVWMvU1R0ZjFIRHBNM2I10TJ5N2gzVGh4NW96Szy5SExwWvd1QXdhcVM1Y3Yy
NnE3Y2Vi0GVmV1lhUmVQM2lGVTh6ajFrb1N3WlhITW1uQ2pZME9nYwxvN1VRZ1NDTTNxUVfYmkgvWEZQ
N3NzWhG0NV1s0TFCeWVDZXA0bW9ab0grMwZHM3hENHRUN3g4a3d5ajhud2I5ZXyN1YwQjZkKzdINHPL
dnVkJUg1MzdGanF5ek9IZEpuSEV1em1YcS9XanhPYnZ0TWJ2N25oeXdzWDJhVnNXdEM4KzQ4YUx1YXBF
N3A1d0taaTBBmkFRU1Y1bnZSNEUrDUpjK2I2MwtBcHFJbnhCZ21kLzRWNVFQL210MThIREM3c1JIZnRt
ZXU1bG1oVjBybi9BTfGyMzJicWQ0QkZuRHg3VmKxY1dTmNvmZjBJYkI0N3FleHhtVWo5UXV0Wwp1cGQz
dF1ENmFiv0JCTXJoK2FwTmJPS3JORjErdWdDYTRYaVhHZndNUFB0Vm1hdmhVM11NT0FBbnVvYi9SMDdM
MH1PU2VPYWRfODhBcHNYRkdmZjMweW5obEpnTTUxQ1U2dk45RXpnbB2SEJGVXlpVnJhZVBpd0o1M0RG
NVpUWm5vbUV0Zzg1a05VZDJvSmkyV3ByNE9tbWtmTjR4NHpIZmlWRmM4RHY4Tnp1aE5xT2lkaWxHdkE2
REd1ZVp3Tzc4QUFRbjZjaUVrNitydzVWY3ZqdnFORf1QT29JVXdhS1NocnhBdVhMbGtINGFZdUdmTV1E
YzEwV0Y1VGEzMWhQSk9mY1Voc1UvSmxJTmk2YzZ1bFJZZEJwbzYrK1lmang2MwxHTmZSbTRNRDVySjFq
M0ZvR0huakRTQk5hc1lVZ01MeU1zektwYjd0WHBvSGZQczhoM1dwMux6TmZOazU0WHhDMXdER1VtWXPY
WWVmaDZ6L2NLdFZtNEVCeGE5V1FHRHpZcJnMc1VNumpIRUtrazd6YUzLWVFBMmhHUVUxeis4NU5GV3BY
RHJrejN2eDEwR3F4UTZCemVOYm9CazVuOGs0bmViUmgrazFoV2Z4VEYwRDFFeVdVczVuditkZ1FxS2F4
enVDZEUwaXNIbDAYt1E4YwgwbVhyMTJMYTntMGY5d2lR0St3TE5UTVkvODZNUG84eWkzMU9meG1UN1BX
b3FH0StEwnVrWw5hNTZtU1p0NVdXU3k1cVZBMXJ3VX1KcVhBbG56a21haS9nSFNEN1JrVHlpaG9nQUFB
QUJKU1U1RXJrSmdnZz09IiwKCQkJCSJzdXBwb3J0ZWRFehR1bnNpb25zIjogW3sKCQkJCQkJIm1kIjog

ImhtYWMtc2VjcmV0IiwKCQkJCQkJImZhaWxfawZfdW5rbm93biI6IGZhbnHNlCgkJCQkJfSwKCQkJCQl7
CgkJCQkJCSJpZCI6ICJjcmVkuUHJvdGVjdCIsCgkJCQkJCSJmYwlsX2lmX3Vua25vd24iOiBmYXxzZQoJ
CQkJCX0KCQkJCv0sCgkJCQkiYXV0aGVudGljYXRvckdlEluZm8iOiB7CgkJCQkJInZlcnNpb25zIjog
WyJVMkZfVjIiLCAiRklET18yXzAiXSwwKCQkJCQkiZXh0ZW5zaW9ucyI6IFsiY3JlZFBYb3RlY3QiLCAi
aG1hYy1zZW5yZXQiXSwwKCQkJCQkiYWFndWlkIjogIjAxMzJkMTEwYmY0ZTQyMDhhNDZyZW50ZjVMTJl
ZmU1IiwKCQkJCQkib3B0aW9ucyI6IHsKCQkJCQkJInBsYXQiOiAiZmFsc2UiLAoJCQkJCQkicmsiOiAi
dHJ1ZSIsCgkJCQkJCSJjbGllbnRQaW4iOiAidHJ1ZSIsCgkJCQkJCSJ1cCI6ICJ0cnVlIiwKCQkJCQkJ
InV2IjogInRydWUiLAoJCQkJCQkidXZUb2t1biI6ICJmYXxzZSIsCgkJCQkJCSJjb25maWciOiAiZmFs
c2UiCgkJCQkJfSwKCQkJCQkibWF4TXNnU2l6ZSI6IDEyMDAsCgkJCQkJInBpb1V2QXV0aFByb3RvY29s
cyI6IFsXSwKCQkJCQkibWF4Q3JlZGVudGlhbENvdW50SW5MaXN0IjogMTYsCgkJCQkJIm1heENyZWRl
bnRpYXwJZExlbmd0aCI6IDEyMDAwCgkJCQkJCQkidHJhbnNwb3J0cyI6IFsidXNiIiwgIm5mYyJdLAoJCQkJ
CSJhbGdvcml0aG1zIjogW3sKCQkJCQkJCSJ0eXB1IjogInB1YmXpYy1rZXkiLAoJCQkJCQkJImFsZyI6
IC03CgkJCQkJCX0sCgkJCQkJCXsKCQkJCQkJCSJ0eXB1IjogInB1YmXpYy1rZXkiLAoJCQkJCQkJImFs
ZyI6IC0yNTcKCQkJCQkJfQoJCQkJCv0sCgkJCQkJIm1heEF1dGhlbnRpY2F0b3JDb25maWdMZW5ndGgi
OiAxMDI0LAoJCQkJCSJkZWZhdWx0Q3JlZFBYb3RlY3QiOiAyLAoJCQkJCSJmaXJtd2FyZVZlcnNpb24i
OiA1CgkJCQl9CgkJCX0sCgkJCSJzdGF0dXNSZXBvcnRzIjogW3sKCQkJCQkic3RhdHVzIjogIkZJRE9f
Q0VSVElGSUVEIiwKCQkJCQkiZWZmZW50aXZlRGF0ZSI6ICJyMDE5LTAxLTAAIgoJCQkJfSwKCQkJCXsK
CQkJCQkic3RhdHVzIjogIkZJRE9fQ0VSVElGSUVEIiwKCQkJCQkiZWZmZW50aXZlRGF0ZSI6ICJy
MDIwLTExLTExIiwKCQkJCQkiY2VydGlmaW50dGhlbnRlc2NyaXB0b3IiOiAiRklETyBBbGxpYw5jZSBT
YW1wbGUgRklETzIgQXV0aGVudGljYXRvciIsCgkJCQkJImNlcnRpZmljYXRlTnVtYmVYIjogIkZJRE8y
MTAwMDIwMTUxMjIwMDAxIiwKCQkJCQkiY2VydGlmaW50dGhlbnRlc2VybGljeVZlcnNpb24iOiA1MS4wLjEi
LAoJCQkJCSJjZXJ0aWZpY2F0aW9uUmVxdWlyZW11bnRzVmVyc2lubiI6ICJlLjAuMSIKCQkJCX0KCQkJ
XSwwKCQkJInRpbWVpZkxhc3R0dGF0dXNDaGFuZ2UiOiAiMjAxOS0wMS0wNCIKCQl9CglldCn0

EXAMPLE: JWT HEADER

```
{
  "alg": "ES256",
  "typ": "JWT",
  "x5c": [
    "MIICZTCCAgugAwIBAgIBATAKBggqhkJOPQQAjCBozEnMCUGA1UEAwweRVhBTBVBMR  

    RSNRFRMzIFRFU1QgSU5URVJNRURJQVRFMStIAYJKoZIhvcNAQkBFhNleGFtcGx1  

    QGV4YW1wbGUuY29tMRQwEgYDVQQKDAFeGFtcGx1IE9SRzEQMA4GA1UECwwHRXhh  

    bXBsZTElMAkGA1UEBhMCVVMxZzA1BjBgNVBAGMAk1ZMRiEAYDVQQHDA1XYWt1Zmll  

    bGQwHhcNMjEwNDE5MTEzNTA3WhcNMjEwNDE5MTEzNTA3WjCBPTEpMCcGA1UEAwg  

    RVhBTBVBMRSNRFRMzIFNJR05JTkcqQ0VSVElGSUNBVEUxIjAgBgkqhkiG9w0BCQEW  

    E2V4YW1wbGV4ZXhhbXBsZS5jb20xZDAsBgNVBAoMC0V4YW1wbGUgT1JHMRAwDgYD  

    VQQLDAdFeGFtcGx1MQswCQYDVQQGEwJVUzELMAkGA1UECAwCTVxkEjAQBGNVBACM  

    CVdha2VmaWVsZDBZBMGBYqGSM49AgEGCCqGSM49AwEHA0IABNqJs6wTqixc+S+V  

    DAajF1PNat10KEWJE5jcW0vm6qp09SDAAMZvb4HHRvs+P5YRpHrSlUPdvK+uEQbd  

    Wg31P9ujLDAqMAkGA1UdEwQCAAwHQYDVR00BBYEFQsapcXV4Z0VHAnRpPZwQe7  

    Yy20MAoGCCqGSM49BAMCA0gAMEUCIQ67za8EIuyRiKgNDXIP1s1aLr3jzH9VWXf  

    Hx4bJ+zCsgIgG/tVButOJUu+vvoHio/otAUACH5bNHP3uIziDS+PTUc=",
    "MIIHhZCCAgagAwIBAgIBAJANBgkqhkiG9w0BAQsFADCBmzEfmB0GA1UEAwWRVhB  

    TVBMRSNRFRMzIFRFU1QgUk9PVDEiMCAGCSqGSIb3DQEJARYTZXhhbXBsZUBleGFt  

    cGx1LmNvbTEUMBIGA1UECgwLRXhhbXBsZSBPUKcxEDA0BgNVBAsMB0V4YW1wbGUx  

    CzA1BjBgNVBAYTA1VTMQswCQYDVQQIDAjNWTESMBAGA1UEBwwJV2FrZWZpZWxkMB4X
```

```
DTIxMDQxOTExMzUwN1oXDTQ4MDkwNDExMzUwN1owgaMxJzAlBgNVBAMMHkVYQU1Q
TEUgTURTMYBURVNUIELOVEVSTUVESUFURTEiMCAGCSqGSIB3DQEJARYTZXhbbXBs
ZUBleGFtcGx1LmNvbTEUMBIGA1UECgwLRXhbbXBsZSBPUkcxEDAOBgNVBAsMB0V4
YW1wbGUxCzAJBgNVBAYTA1VTMQswCQYDVQQIDAjNWTESMBAGA1UEBwwJV2FrZWZp
ZWxkMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAENGumBbYnFQnTjP1RSfc70hsh
gbiI1ZtpwQ5n6xRLA/Wq0PSCfL15qQ+r7d1cK1d3r3vLa+vm6G6vKHGCPeEuzqMv
MC0wDAYDVR0TBAUwAwEB/zAdBgNVHQ4EFgQUK6F4RjNggVFe+0/cbZwfrZd7ZUw
DQYJKoZIhvcNAQELBQADggIBACnp1fm0FKlWmUtTp1LuYg7mps4xP/C0u8dnb38u
1nMDVu0T4+CZaiM9AGz313GD22hjLGrmPuYn86wGOKI3H0rEpsGdMmfy7tTmKX/e
M/eS3FEDXZnE82Pn5oFIyBT/f8sGuXyOsFZqWBvVdBIIDldCpD4mxMQZZOZtTrlv
3WvBQMC/dsicOxe3QKXvWHi6Qb/Rhuaip3rPmwMf+4JpnJO+JMPqAaU1cAH8HVsf
rLAMoKs148j2+cvbpawmsT5rIoH/ezVrPaG/M0iIgq79w/efuvSi5AX8J+kDoLSE
f3d5w0gkJYAqUqcRxXTEETKIzDM6hzaBQFiAWvTn9IlVWgntQamSXvH+txaTF9iE
lHxUf5INYFVciCpztSrydeHv/OCNRF7/LVricMSlo8Rh+03yP9V+2uNf3X8sQJNt
ufrQNaqq18wiXliTLufSn02/g+mkhIUiNKfT0JpvCjKeCnCFcxQU2/XT3Kh3G8gD
Jws06EVRjMUJt4AYKze/hEUCwF55IF2m3jHIoCu8jVfj24CeEX5dnfvSr+SVvN5Q
B0uZ05M4rmyZXyqBm0zK3fR+iE0/ZpInuwLC7X+W82zXlnMkplI3Q+Jxd7jfQ15S
YNE2K6rvRIT01w0P9ZqyDF7knGKpRlp70qxd37bD/VUbWpQ7gIAfsJNH5KBLowHJ
FFjw"
```

]

}

In order to produce the tbsPayload, we first need the base64url-encoded (without padding) JWT Header:

EXAMPLE: ENCODED JWT HEADER

```
ewogICJhbGciOiAiAUMyNTYiLAogICJ0eXAiOiAiSlldUiwKICAieDVjIjogWwogICAgIk1JSUNaVEND
QWd1Z0F3SUJBZ0lCQVRBS0JnZ3Foa2pPUFFFRREFqQ0JvekVuTUNVR0ExVUVBd3d1UlZoQ1RWQk1SU0JO
UkZNeK1GUkZVMVFuU1U1VWJWSk5SVVJKUVZSRk1TSXDJQVlKS29aSWh2Y05BUWtCRmhObGVHRnRjR3hs
UUDWNFlMXMxd1R1V1WTI5dE1SUxZlE1EVlFRS0RBdE1R0Z0Y0d4bE1FOVNSekVRTUE0R0ExVUVBd3d1
UlhoaGJYQnNaVEVMTUFR0ExVUVCaE1DVlZNeEN6QUUpCZ05WQkFnTUFRmVpNUk13RUFZRFZRUUHEQWxY
WVd0bFptbGxiR1F3SGhjTk1qRXd0REU1TVRFek5UQTNXaGNOTXpFd05ERTNNVEV6TlRBM1dqQ0JwVEVw
TUNjR0ExVUVBd3d1UlZoQ1RWQk1SU0JOuKZNeK1GTkpSMDVKVGtjZ1EwVlNWRWxHU1V0Q1ZlFVXhJakFn
QmdrcWhraUc5dzBCQ1FFV0UyYjRZVzF3YkdWQVpYaGhiWEJzWlM1amIyMHhGREFTQmdOVk1Bb01DMFY0
WVcxZD2JHVWdUMUpITVJBd0RnWURWUWVFMREFkRmVHRnRjR3hsTVFzd0NRWURWUWFHRXdKVlV6RUxNQWtH
QTFVRUNBd0NUVmt4RwPBUUJnTlZCQWNNQ1ZkaGEyVm1hV1ZzWkRCWk1CTUdCeXFHU000OUFnRUdDQ3FH
U000OUF3RUhBME1BQk5RSnM2d1RxaXhjK1MrVkrBYWpGbFBOYXQxMEtFV0pFNWpjv092bTZxcE85U0RB
QU1admI0SEhydnMrUDVZUnBic1NsVVBkdksrdUVRyMRXZzMXUD11akxEQXFNQWtHQTFVZEV3UUNNQUF3
SFFZRFZSMSE9CQ1lFRkxzc2FwY1hWNFpVvKhBb1JwUFP3UWU3WXYME1Bb0dDQ3FHU000OUJBTUNBMGdB
TUVVQ01RQzY3emE4RU11eVJpS2d0RFhJUDFzMWFMcjNqekg5V1ZYZkh4NGJKK3pDc2dJZ0cvdFZCdXRP
S1VVK3Z2b0hJby9vdEFVQWNINWJOSFAzdU16aURTK1BUVWM9IiwKICAieDVjIjogWwogICAgIk1JSUNaVEND
QWd1Z0F3SUJBZ0lCQVRBS0JnZ3Foa2pPUFFFRREFqQ0JvekVuTUNVR0ExVUVBd3d1UlZoQ1RWQk1SU0JO
UkZNeK1GUkZVMVFuU1U1VWJWSk5SVVJKUVZSRk1TSXDJQVlKS29aSWh2Y05BUWtCRmhObGVHRnRjR3hs
UUDWNFlMXMxd1R1V1WTI5dE1SUxZlE1EVlFRS0RBdE1R0Z0Y0d4bE1FOVNSekVRTUE0R0ExVUVBd3d1
UlhoaGJYQnNaVEVMTUFR0ExVUVCaE1DVlZNeEN6QUUpCZ05WQkFnTUFRmVpNUk13RUFZRFZRUUHEQWxY
WVd0bFptbGxiR1F3SGhjTk1qRXd0REU1TVRFek5UQTNXaGNOTXpFd05ERTNNVEV6TlRBM1dqQ0JwVEVw
TUNjR0ExVUVBd3d1UlZoQ1RWQk1SU0JOuKZNeK1GTkpSMDVKVGtjZ1EwVlNWRWxHU1V0Q1ZlFVXhJakFn
QmdrcWhraUc5dzBCQ1FFV0UyYjRZVzF3YkdWQVpYaGhiWEJzWlM1amIyMHhGREFTQmdOVk1Bb01DMFY0
WVcxZD2JHVWdUMUpITVJBd0RnWURWUWVFMREFkRmVHRnRjR3hsTVFzd0NRWURWUWFHRXdKVlV6RUxNQWtH
QTFVRUNBd0NUVmt4RwPBUUJnTlZCQWNNQ1ZkaGEyVm1hV1ZzWkRCWk1CTUdCeXFHU000OUFnRUdDQ3FH
U000OUF3RUhBME1BQk5RSnM2d1RxaXhjK1MrVkrBYWpGbFBOYXQxMEtFV0pFNWpjv092bTZxcE85U0RB
QU1admI0SEhydnMrUDVZUnBic1NsVVBkdksrdUVRyMRXZzMXUD11akxEQXFNQWtHQTFVZEV3UUNNQUF3
SFFZRFZSMSE9CQ1lFRkxzc2FwY1hWNFpVvKhBb1JwUFP3UWU3WXYME1Bb0dDQ3FHU000OUJBTUNBMGdB
TUVVQ01RQzY3emE4RU11eVJpS2d0RFhJUDFzMWFMcjNqekg5V1ZYZkh4NGJKK3pDc2dJZ0cvdFZCdXRP
S1VVK3Z2b0hJby9vdEFVQWNINWJOSFAzdU16aURTK1BUVWM9IiwKICAieDVjIjogWwogICAgIk1JSUNaVEND
QWd1Z0F3SUJBZ0lCQVRBS0JnZ3Foa2pPUFFFRREFqQ0JvekVuTUNVR0ExVUVBd3d1UlZoQ1RWQk1SU0JO
UkZNeK1GUkZVMVFuU1U1VWJWSk5SVVJKUVZSRk1TSXDJQVlKS29aSWh2Y05BUWtCRmhObGVHRnRjR3hs
UUDWNFlMXMxd1R1V1WTI5dE1SUxZlE1EVlFRS0RBdE1R0Z0Y0d4bE1FOVNSekVRTUE0R0ExVUVBd3d1
UlhoaGJYQnNaVEVMTUFR0ExVUVCaE1DVlZNeEN6QUUpCZ05WQkFnTUFRmVpNUk13RUFZRFZRUUHEQWxY
WVd0bFptbGxiR1F3SGhjTk1qRXd0REU1TVRFek5UQTNXaGNOTXpFd05ERTNNVEV6TlRBM1dqQ0JwVEVw
TUNjR0ExVUVBd3d1UlZoQ1RWQk1SU0JOuKZNeK1GTkpSMDVKVGtjZ1EwVlNWRWxHU1V0Q1ZlFVXhJakFn
QmdrcWhraUc5dzBCQ1FFV0UyYjRZVzF3YkdWQVpYaGhiWEJzWlM1amIyMHhGREFTQmdOVk1Bb01DMFY0
WVcxZD2JHVWdUMUpITVJBd0RnWURWUWVFMREFkRmVHRnRjR3hsTVFzd0NRWURWUWFHRXdKVlV6RUxNQWtH
RFRJeE1EUXhPVEV4TXpVd04xb1hEVFE0TURrd05ERXhNe1V3TjFvd2dhTXhKekFsQmdOVk1Bb01DMFY0
WVcxZD2JHVWdUMUpITVJBd0RnWURWUWVFMREFkRmVHRnRjR3hsTVFzd0NRWURWUWFHRXdKVlV6RUxNQWtH
```

UVUxUVRfVwdUVVJUTXlCVVJwTlVJRwXPVkvWU1RVVkvTVUzVU1RFaU1DQUdDU3FHU01iM0RRRUpBU1lU
WlhoaGJYQnNaVUJszUdGdGNHeGxMbU52YlRFVU1CSUdBMVVFQ2d3TFJYaGhiWEJzWlNCUFVrY3hFREFP
QmdOVkJBc01CMFY0WVcxdb2JHVXhDekFKQmdOVkJBWVRBbFZUTVfZd0NRWURWUVFJREFKtldURVNNQkFH
QTFVRUJ3d0pWmkZyWldacFpXEGtNRmt3RXdZSEtVwk16ajBDQVFZSutVwk16ajBEQVFjRFFnQUVOR3Vt
QmJZbkZrBlRqUDFSU2ZjNzBoc2hnYm1JMVp0cHdRNW42eFJMqS9XcTBQU0NmTGw1cVErcjdbkGNLMWQz
cjN2TGERdm02RzZ2S0hHQ1BFZV6cU12TUMwd0RBWURWUjBUQkFVd0F3RUIvekFkQmdOVkhRNEVGZ1FV
Tms2RjRssm5HR1ZGZSSwL2NiWndmc1pkN1pVd0RRWUplb1pJaHZjTkFRRUxCUUFEZ2dJQkFDbnAxZm0w
RktsV21VdFRwbEx1Wwc3bXBzNHhQL0NPdThkbmIzOHUxbk1EVnVPVDQrQ1phaU05QUd6MzEzR0QyMmhq
TEdybVB1WW44NndHT0tJM0hPckVwc0dkTW1meTd0VG1LWC91TS91UzNGRURYWm5FODJQbjVvRk15Q1Qv
ZjhzR3VYeU9zRlpxV0J2VmRCSU1EbGRDcEQ0bXhNUVpaT1p0VHJsdjNXdkJRTUMvZHNpY094ZTNRS1h2
V0hpNlFiL1JodWFpcDNyUG13TWYrNEpwbkPK0pNUHfBYVUxY0FIOEhWc2ZyTEFNb0tzMTQ4ajIrY3Zi
cGFxbXNUNXJJb0gVZXPwc1BhRy9NT2lJZ3E3OXcvZWZ1d1NpNUFYOEora0RvTFNFZjNkNXdPZ2tKWUFx
VXFjUnhYVEVfDEtJekRNNmh6YUJRrm1BV3ZUbjlJbFZXZ250UWFtU1h2Sct0eGFURjlpRwxIeFVmNU10
WUZWY2lDcHp0U3J5ZGVIdi9PQ05SZjcvTFZyaWNNU2xvOFJoK08zeVA5VisydU5mM1g4c1FKTnR1ZnJR
TmFxcTE4d2lYbGlUTHVmU24wMi9nK21raElVaU5LZlRPSnB2Q2pLZUNuQ0ZjeFFVMi9YVDNLadNHOGdE
SndzTzZfVlJqTVVKdDRBWUt6ZS9oRVVDd0Y1NU1GMM0zakHjB0N10GpWZmoyNEN1RVg1ZG5md1NyK1NW
dk41UUIdVowNU00cm15Wlh5cUJtMHPLM2ZSK2lFMC9acEludXdmQzdYK1c4MnpYbG5Na3BsSTNRK0p4
ZDdqZlExNVNZTKUySzZyd1JJVDAXdzBQOVpxeURGN2tuR0twUmxwN09xeGQzN2JEL1ZVYldwUTdnSUFm
c0pOSDVLQkxvd0hKRkZqVYIKICBdCn0

then we have to append a period (".") and the base64url encoding of the EncodedMetadataBLOBPayload (taken from the example in section [Metadata BLOB Format](#)):

EXAMPLE: TBSPAYLOAD

eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsIng1YyI6WyJNSU1DWlRDQ0FndWdBd01CQWdJQkFUQUtC
Z2dxaGtqT1BRUURBakNCb3pfbk1DVUdBMVVFQXd3ZVJWaEJUUVk1JNU1NCTlJGTxpJRlJGVTFRZ1NVNVVS
VkpOU1VSS1FWUKZNU013SUFZSkTVWkldmNOQVFRkZ0Tmx1R0Z0Y0d4bFFHVjRZVzF3YkdVdVkyOXRN
U1F3RwdZRFZRUUtEQXRGZUdGdGNHeGxJRTlTUnpFUU1BNEdbMVVFQ3d3SFJYaGhiWEJzWlRFTE1Ba0dB
MVVFQmhnQ1ZWtXhDekFKQmdOVkJBZ01BazFaTVJJd0VBWURWUVFIREFsWFlXdGxabWxsYkdRd0hoY05N
akV3TkRfNU1URXpOVEEzV2hjTk16RXdOREUzTVRFek5UQTNXakNCcFRFcE1DY0dBWVVFQXd3Z1JWaEJU
Vk1JNU1NCTlJGTxpJRk5KUjA1SlRrY2dRMFZTVkVsR1NVtkJWRVV4SWpBZ0Jna3Foa2lHOXcwQkNRRVdF
MlY0WVcxdb2JHVkFawGhoYlhCc1pTNWpiMjB4RkrBU0JnTlZCQW9NQzBWNFlXMxd1R1VnVDFKSE1SQXdE
Z11EV1FRTERBZEZ1R0Z0Y0d4bE1Rc3dDUV1EV1FRR0V3SlZVekVMTUFR0ExVUVDQXdDVfZreEVqQVFC
Z05WQkFjTUNWZGhhM1ZtYVdwc1pEQ1pNQk1HQnlxR1NNND1BZ0VHQ0NxR1NNND1Bd0VIQTBJQUJOUpz
NndUcWl4YytTK1ZEQWFqRmxQTmF0MTBLRVdKRTVqY1dPdm02cXBPOVNEQUFNWnZiNEhIcnZzK1A1WVJw
SHJTBfVQZHZLK3VFUWJkV2czMVA5dWpMREfXTUFR0ExVWRfD1FDTUfBd0hRWURWUjBPQkJZRUZMcXNh
cGNyVjRab1ZlQW5ScFBad1FlN1l5MjBNQW9HQ0NxR1NNND1CQU1DQTBnQU1FVUNJUUM2N3ph0EVJdXlS
aUtnTKRYSVAxczFhTHIzanpIOvdWwGZiEDrISit6Q3NnSwdHL3RWQnV0T0pVVSt2dm9ISW8vb3RBVUFj
SDViTkHQM3VJem1EUytQVFVjPSIsIk1JSUVIEkNDQWdlZ0F3SUJBZ01CQWpBTk1na3Foa2lHOXcwQkFR
c0ZBRENCbXpFZk1CMEdBMVVFQXd3V1JWaEJUUVk1JNU1NCTlJGTxpJRlJGVTFRZ1Vr0VBWREVpTUNBR0NT
cUdTSWIZRFFFSkFSWVRawGhoYlhCc1pVQmx1R0Z0Y0d4bExtTnZiVEVVTUJJR0ExVUVDZ3dMUlhoaGJY
QnNaU0JQVWtjeEVEQU9CZ05WQkFzTUIwVjRZVzF3YkdVeEN6QU9CZ05WQkFZVEFsV1RNUNXN3Q1FZRFZR
UUEQU9pOV1RFU01CQUdBMVVFQnd3SlYyRnJav1pwWld4a01CNFhEVEl4TURReE9URXhNe1V3TjFvWERU
UTRNRGt3TkRFeE16VXdOMW93Z2FNeEp6QWx0Z05WQkFNTUhrV1lRVTFRVEVZ1RVU1RNeUJVVU1ZOVU1F
bE9WRVZTVFVWRVNV1VSVEVpTUNBR0NTcUdTSWIZRFFFSkFSWVRawGhoYlhCc1pVQmx1R0Z0Y0d4bExt

TnZiIVEVVTUJJR0EXVUVDZ3dMUlhoaGJYQnNaU0JQVWtjeEVEQU9CZ05WQkFzTUIWvjRZVzF3YkdVeEN6
QUpCZ05WQkFZVEFsVlRNUXN3Q1FZRFZRUU1EQUpOV1RFU01CQUdBMVVFQnd3S1YyRnJaV1pwWld4a01G
a3dFd1lIS29aSXpqMENBUVlJS29aSXpqMERBUWNEUWdBRU5HdW1CY1luRlFuVgPQMvJTZmM3MGhzaGdi
aUkxWnRwd1E1bjZ4UkxBL1dxMFBTQ2ZMbDVxUSTyN2RsY0sxZDNyM3ZMYSt2bTZHNnZLSEdDUEVlVXpx
TXZNQzB3REFZRFZSMFRCQVV3QXdfQi96QWRCZ05WSFE0RUZnUVV0azZGNFJKbkdhVkJkZ1KzAvY2Jad2Zy
WmQ3WlV3RFFZSkTvWklodmNOQVFFTEJRQURnZ01CQUdNucDFmbTBGS2xXbVV0VHBsTHVZZzdtcHM0eFAv
Q0910GRuYjM4dTfUTURWdU9UNCtDwmFpTTlBR3ozMTNHRDIyaGpMR3JtUHVZbjg2d0dPS0kzSE9yRXBz
R2RnbWZ5N3RUBUtYL2VNL2VTM0ZFRFhabkU4M1BuNW9GSX1CVC9mOHNHdVh5T3NGWnFXQnZWZEJJSURS
ZENwRDRteE1RWlPpWnRUcmx2M1d2Q1FNQy9kc21jT3h1M1FLWHZXSgk2UWivUmh1Yw1wM3JQbXdnZis0
SnBuSk8rSk1QcUFhVTFjQUg4SFZzZnJMQU1vS3MxNDhqMitjdmJwYVdtc1Q1cklvSC91elZyUGFHL01P
aU1ncTc5dy91ZnV2U2k1QVg4SitrRG9MU0VmM2Q1d09na0pZQXFVcWNSeFhURUV0S016RE02aHphQ1FG
aUFXd1RuOU1sVldnbnRRYw1TWHZIK3R4YVRGOW1FbEh4VWY1SU5ZR1ZjaUNwenRTcnlkZUh2L09DT1Jm
Ny9MvnJpY01TbG84UmgrTzN5UDlWKzJ1TmYzWdhzUUOpdHVMc1FOYXFxMTh3aVhsaVRMDwZTbjAyL2cr
bWtoSVVpTktmVE9KcHZDaktlQ25DRmN4UVUyL1hUM0toM0c4Z0RKd3NPNkVWUmpNVUp0NEFZS3p1L2hF
VUN3RjU1SUyYbTNqSElvQ3U4a1ZmajI0Q2VFWDVkbmZ2U3IrU1Z2TjVRQjB1WjA1TTRYbX1aWH1xQm0w
ekszZ1IraUUwL1pwSW51d0xDN1grVzgyelhsbk1rcGxJM1ErSnhkN2pmUTE1U11ORTJLnnJ2Uk1UMDF3
MFA5WnF5REY3a25HS3BSbHA3T3F4ZDM3YkQvV1ViV3BRN2dJQWZzSk5INUtCTG93SEpGRmpXI119.eyJ
sZWdhbEh1YWRLciI6IlJldHJpZXZhbCBhbmQgdXNlIG9mIHRoaXMgQkxPQiBpbmRPy2F0ZXMGyWNjZXB
0YW5jZSBvZiB0aGUgYXBwcm9wcm1hdGUgYWdyZWVtZW50IGxvY2F0ZWQgYXQgaHR0cHM6Ly9maWRvYXw
saWFuY2Uub3JnL21ldGFkYXRhL21ldGFkYXRhLWxlZ2F5LXRLcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlb3RyZWVzIjpbeyJhYWlkIjoiaMTIzNCM1Njc4IiwibWV0YWRhdGFtdGF
0ZW1lbnQiOnsibGVnYWxzIjZWFkZXIiOiI0JodHRwc3ovL2ZpZG9hbGxpYW5jZS5vcmcvbwV0YWRhdGEvbwV
0YWRhdGEtc3RhdGVtZW50LWxlZ2F5LWwh1YWRLci8iLCJkZXNjcmlwdGlvbiI6IkZJRE8gQWxsawFuY2U
gU2FtcGx1IFVBRiBBdXRoZW50aWNhdG9yIiw1YWFpZCI6IjEyMzQjNTY3OCIsImFsZGVybWw0aXZlRGV
yY3JpcHRpb25zIjpb7InJ1LVJVIjojIj_RgNC40LzQtdGAIQVBRiDQsNGD0YLQtdC90YLQuNGE0LjQutC
w0YLQvtGA0LAG0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlRlIiOiJFeGvtcGx1IFVBRiBhdXRoZW50aWN
hdG9yIGRlIEZJRE8gQWxsawFuY2UifSwiYXV0aGVudGljYXRvc1ZlcnNpb24iOjIsInByb3RvY29sRmF
taWw5IjoidWFmIiwic2NoZW1hIjozLCJ1cHYiOi0lt7Im1ham9yIjoxLCJtaW5vciI6MmH0seyJtYWpvcii
6MSwibWlub3IiOjF9XSwiYXV0aGVudGljYXRpb25BbGdvcml0aG1zIjpbInNlY3AyNTZyMv91Y2RzYV9
zaGEyNTZfcml0aG1zIjpbInB1Ym9yY0tleUFSZ0FuZEVuY29kaw5ncyI6WYj1Y2NfeDk2M19yYXciXSwiYXR
0ZXN0YXRpb25UeXB1cyI6WYj1YXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcm1maWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaw50X2ludGVybWw0aXZlRGVzIiw1YmFEZXNjIjpb7InN
lbGZBdHRlc3RlZEZBUiI6MC4wMDAwMiwibWw4dFVwZGF0aG1zIjpbInNlY3AyNTZyMv91Y2RzYV9
zaGEyNTZfcml0aG1zIjpbInB1Ym9yY0tleUFSZ0FuZEVuY29kaw5ncyI6WYj1Y2NfeDk2M19yYXciXSwiYXR
0ZXN0YXRpb25UeXB1cyI6WYj1YXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcm1maWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaw50X2ludGVybWw0aXZlRGVzIiw1YmFEZXNjIjpb7InN
lbGZBdHRlc3RlZEZBUiI6MC4wMDAwMiwibWw4dFVwZGF0aG1zIjpbInNlY3AyNTZyMv91Y2RzYV9
zaGEyNTZfcml0aG1zIjpbInB1Ym9yY0tleUFSZ0FuZEVuY29kaw5ncyI6WYj1Y2NfeDk2M19yYXciXSwiYXR
0ZXN0YXRpb25UeXB1cyI6WYj1YXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcm1maWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaw50X2ludGVybWw0aXZlRGVzIiw1YmFEZXNjIjpb7InN
lbGZBdHRlc3RlZEZBUiI6MC4wMDAwMiwibWw4dFVwZGF0aG1zIjpbInNlY3AyNTZyMv91Y2RzYV9
zaGEyNTZfcml0aG1zIjpbInB1Ym9yY0tleUFSZ0FuZEVuY29kaw5ncyI6WYj1Y2NfeDk2M19yYXciXSwiYXR
0ZXN0YXRpb25UeXB1cyI6WYj1YXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcm1maWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaw50X2ludGVybWw0aXZlRGVzIiw1YmFEZXNjIjpb7InN
lbGZBdHRlc3RlZEZBUiI6MC4wMDAwMiwibWw4dFVwZGF0aG1zIjpbInNlY3AyNTZyMv91Y2RzYV9
zaGEyNTZfcml0aG1zIjpbInB1Ym9yY0tleUFSZ0FuZEVuY29kaw5ncyI6WYj1Y2NfeDk2M19yYXciXSwiYXR
0ZXN0YXRpb25UeXB1cyI6WYj1YXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcm1maWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaw50X2ludGVybWw0aXZlRGVzIiw1YmFEZXNjIjpb7InN
lbGZBdHRlc3RlZEZBUiI6MC4wMDAwMiwibWw4dFVwZGF0aG1zIjpbInNlY3AyNTZyMv91Y2RzYV9
zaGEyNTZfcml0aG1zIjpbInB1Ym9yY0tleUFSZ0FuZEVuY29kaw5ncyI6WYj1Y2NfeDk2M19yYXciXSwiYXR
0ZXN0YXRpb25UeXB1cyI6WYj1YXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcm1maWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaw50X2ludGVybWw0aXZlRGVzIiw1YmFEZXNjIjpb7InN
lbGZBdHRlc3RlZEZBUiI6MC4wMDAwMiwibWw4dFVwZGF0aG1zIjpbInNlY3AyNTZyMv91Y2RzYV9
zaGEyNTZfcml0aG1zIjpbInB1Ym9yY0tleUFSZ0FuZEVuY29kaw5ncyI6WYj1Y2NfeDk2M19yYXciXSwiYXR
0ZXN0YXRpb25UeXB1cyI6WYj1YXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcm1maWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaw50X2ludGVybWw0aXZlRGVzIiw1YmFEZXNjIjpb7InN
lbGZBdHRlc3RlZEZBUiI6MC4wMDAwMiwibWw4dFVwZGF0aG1zIjpbInNlY3AyNTZyMv91Y2RzYV9
zaGEyNTZfcml0aG1zIjpbInB1Ym9yY0tleUFSZ0FuZEVuY29kaw5ncyI6WYj1Y2NfeDk2M19yYXciXSwiYXR
0ZXN0YXRpb25UeXB1cyI6WYj1YXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcm1maWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaw50X2ludGVybWw0aXZlRGVzIiw1YmFEZXNjIjpb7InN
lbGZBdHRlc3RlZEZBUiI6MC4wMDAwMiwibWw4dFVwZGF0aG1zIjpbInNlY3AyNTZyMv91Y2RzYV9
zaGEyNTZfcml0aG1zIjpbInB1Ym9yY0tleUFSZ0FuZEVuY29kaw5ncyI6WYj1Y2NfeDk2M19yYXciXSwiYXR
0ZXN0YXRpb25UeXB1cyI6WYj1YXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcm1maWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaw50X2ludGVybWw0aXZlRGVzIiw1YmFEZXNjIjpb7InN
lbGZBdHRlc3RlZEZBUiI6MC4wMDAwMiwibWw4dFVwZGF0aG1zIj

DUTBFeEN6QUpCZ05WQkFZVEFsVlRNRmt3RXdZSEtVwKl6ajBDQVFZSUtVwKl6ajBEQVFjRFFnQUVIOGH
2MkQwSFhhNTkvQm1wUTdSWmVoTC9GTUd6RmQxUUJnOXZBVXBPWjNham51UTk0UFI3YU16SDMzb1VTQnI
4ZkhZRHJxT0JiNThweEdxSEpSeVgvNk5RTUU0d0hRWURWUjBPQkJZRUZQb0hBM0NMaHhGyKmwSXQ3ekU
0dzhoazVFSi9NQjhHQTfVZE13UVlNQmFBRlBvSEeZQ0xoeEziQzBJdDd6RTR30GhrNUVKL01Bd0dBMVV
kRXdRRk1BTUJBZjh3Q2dZSUtVwKl6ajBFQXdJRFNBQXdSUUloQUowNlFTWHQ5aWhJYkVLWUtJanNQa3J
pVmRMSWd0ZnNiRFN1N0VySmZ6cjRBAUjxb1lDwMwYk3pJNTVhUWVBSGpJekE5WG02M3JydUF4Ql05cHM
5ejJYTmxRPT0iXSwiaWNvbiI6ImRhDGE6aW1hZ2UvcG5n02Jhc2U2NCxpVkJPUncwS0dnb0FBQUFOU1V
oRVVnQUFBRTbBQUFBdkNBWUFBQUUNpd0pmY0FBQUFBWE5TUjBJQXJzNGM2UUFBUQFSblFVMUJBQU4and
2OF1RVUFBQUFKY0VoWmN3QUFEc01BQUE3REFjZHxR1FBQUFhaFNVUkJWR2hEN1pyNWJ4UmXHTWY5S3p
UQjhBTS9ZRWhFMlC3cFFaY1dLS0JjbFNwSEFubEVMQVJFN2tORUNDQTNGa1dLMENLS1NDRklzS0JjZ1Z
DRFdHTkVTZEFZaWR3Z2dnSkJpUm1NaEZjLzR3eTg4ODR6dTlOZGxuR1RmWkpQmM4zbk8rKzg40TMzZnZ
lQkJ4K1BxQ3pKa1RVdkJiTG1wVURXdkJUSW1wY0NTWnZYTENkWDlSMDVTazE5YmI1YXRmNTk5ZkcrL2V
yQTU0MXE0N2FQMUMVMmE5U015Vk5VaThJaThkNwTHVHNpMzBORnY3Ywk5bjdRWlBNd2JkeXMyZXJVMlH
NcVVkeTgrWmNhTm1HaW1FOHlYTjNSVWQzYTE4bkYwZlVsb3ZaKzBDVHpXcGQyVmorZU9tMWJFeXk2RHg
0aTVwVU1HV3Z1bzUwNnEyMjdkdHVXQk1lZmZyNm9XcFYwRlBOTGhvdzE3NTF0bTIxTHZQSDNyVnRXamZ
6NjZMznFsOHRYN0ZSbd1ZRlNYc21Tc2Vi0WNlT0diWws3TU5VY0dQZzhac2JNZTlyZlFVYFWL0pNWDl
zcWR6RENTdnAwa1pIbVRaZz14N2JMSGNnb1RoYjE2ZUorbVZmUXE4eWfVWlFORzY0aVhaKzAva3E2dU9
aRk8wUXRhDGRXS2ZYb1JROTlCajkxUjVPSUZUazU0ak4wbWtVaXFsTzNYRFcrTWwr0ThtS0I2dFc3cld
wWmNQYyswemc0dExyWwVYzg2RTZlR0RqSU11YlZwY3VzZWfYzmdJWUdSazZicmhaVnIvSmNIem9vTDc
1NTBqZWRRMXhvcFdjQXBpMlpVcWh1N0pMdnJWc1FVODF6a3pPUGVlbU1SWXZWdVfzWDdQYm1EUVK1SnZ
ab25mdEsrMVZZOEg5dXR4NTMwaDBvYitqbVJZcwo2b3VhWXZFZW5XL1dsWwPwOGN3Yk1tNjgydFB3cVc
xUjR0ai8yU0gxM0lSS1lsNG1vWnZyCGlTcURyN2RYdFFIEGEvUESzLytCV3NLMWRUZ0h1N1Y4dFFKM2J
3Rmt3cEzyVU9RNTBzMXIzbGV2bTh6WmNxMTcrQkJhdzdLOGxFSzVxemtZZWFyaz1BOHA3UDNHekRLK25
kM0RRb3crN1VDOFNWTjgyaXV2MzhpbTdOdGFYdFYxQ1ZxNlJndzRwa3NtYmRpM2J1MkRlN1lmYUJCeGN
xZnZxUHJVakZRTlRRMjJsZmRVVlZUNjhYVePLRjVEblNtVWpnZHFNG1TUz1wbXNmREpSM0c2VG9IMG1
XOWFWN0xXTEhZWEtsbFREDDBMVEF0a1lJYWFtcDFRa1Z2Kyt1eUdVeFZkSjBETlZYU20rYjFfxUnhwbDg
0ZGRmWDFMcDFPL2Q20XRzb2QwdnM1aEdyZTl4dThvK2ZwTFIxY0doTlRENlo1N0M5S01XWGVmSmRPWjk
0YmI5b3FkMVJPblM3cUlUvHpIaw1NcWl2Yk8zZzBEZFZ5azNXUUJoQnp0SzM1WUt0ZE9uYzhPM2fjUzZ
mRfPgZ0thWExzRUUpwNXJkcmxpQnFwODljSmNzL203VHZzMHJrakdmTjRiMGtQb1puM1VKdUlpCm5aMjJ
5UDFmbXZVeCtPNWdTcWViVjFtK3pTdVl0VmhxN1RXYkRpTFZ2bGpwbExsb3A2Q0xYUCsycXR2R0xJTC8
xdmltSVNkTUJnelNvRlp5dTZUCwQranp4Z3NQYVY5QkNxZWUvTmPZazZ2NmXL0WN3aVvJl1NUdGYxSER
wTTNiNTkyeTdoM1RoeDVveks20UhmCF1XdUF3YXFTNWN2MjZxN2NlYjh1ZlZZYVJlUDNpRlU4emoxa25
Td1pYSE1tbkNqWTBPZ2FsbzdVUWZTQ00zcVFRcjJIL1hGUDdzc1h4NDVZbDkxQnllQ2VwNG1vWm9IKzF
mRzN4RDR0VDd4OGt3eWo4bndiOWV2MjZWMEI2ZCs3SDR6S3Z1ZEFINTM3RmpxeXpPSGRKbkhdXptWHE
vV2p4T2J2Tk1idjduaHl3c1gyYVZzV3RDOCs00GFMZWfWRTdwnXDLWmkwQTJBUVJWNW52UjRfK3VKYyt
iNjFrQXBxSW54QmdtZC80VjVRUC9tdDE4SERDN3NSSGZ0bWV1NWxtaFYwcm4vQUxYmJMyYnFkNEJGbkR
4N1ZpMwNXUzJ1ZmYwSWJCNDdxZXh4bVVqOVF1dFlqdxBkM3RZRDZhYldCQk1yaCthcE5iT0tyTkYxK3V
nQ2E0cm1YR2Z3TVBQdFZpYXZoVTNZTU9BQW51VwIvUjA3TDB5T1N1T2FkRTg4QXBzWEZHmYzMHlUaGx
KZ001MUNVNnZ0OUV6Z25wdkhCR1V5aVZYyVWQaXdKNTNERjVaVfpub21FTmc4Nwt0VWQyb0ppMldwcjR
Pbw1rZk40eDR6SGZpVkJzjOER2OE56dWhOcU9pZGlsR3ZBNkRHdWVad083OEFBUW42Y2lFazYrcnc1VmN
2anZxTkRZUE9vSVV3YUtTaHJ4QXVYTGxrSDRhWxVHZk1ZRGmXMFdGNVRhMzFoUEpPZmNVaHJVL0psSU5
pNmM2ZWxSWWRCcG82KytZZmp4NjFsR05mUm00TUQ1ckoxajNgB0dIbmpEU0JOYXJZVWdNTH1Nc3pLcGI
3dFhwb0hmUHM4aDNXcDFMek5mTms1NFh4QzF3REdVbVl6WF1lZmg2ei9jS3RwbTRFQnhhOVZRR0R6WXI
zTHJVTVJqSEVL2a3semFGS1lRQTJoR1FVMXorODVORldwWERYa3ozdngxMEdxeFE2Qnp1TmJvQms1bjh
rNG5lYlJoK2sxaFdmeFRGMEQxRXlXVXM1bnYrZGdRcUtheHp1Q2RFMGlzSGwwMk5ROGFoMG1YcjEyTGE
zbTBmOXdpazkrD0xOVE1ZLzg2TVBvOHlpMzFPZnhtVDZQV29xRzkrRfp1a1luYTU2bVNadDVXV1N5NXF

WQTFydc1V5SnFYQWxuemptpYwkvZ0hTRDDSa1R5aWhvZ0FBQFUCFS1JVNUVya0pnZ2c9PSJ9LCJzdGF0dXN
SZXBvcnRzIjpbeyJzdGF0dXMiOiJGSURPX0NFU1RJRklFRClSImVmZmVjdG12ZURhdGUiOiIyMDE0LTA
xLTA0In1dLCJ0aW11T2ZMYXN0U3RhhdHVzQ2hhbmdlIjoImjAxNC0wMS0wNCJ9LHsiYWFnZWlkIjoImDE
zMmQxMTAtYmY0ZS00MjA4LWE0MDMTyWl0ZjVmMTJlZmU1IiwibWV0YWRhdGFTdGF0ZW1lbnQiOnsibGV
nYWxIZWfkZXIiOiJodHRwczovL2ZpZG9hbGxpYW5jZS5vcmcvbwV0YWRhdGEvbWV0YWRhdGEtc3RhZGV
tZW50LWxlZ2FzLWlhYWRLci8iLlCjKjXZNjcmlwdGlvbiI6IkZJRE8gQWxsawFuY2UgU2FtcGx1IEZJRE8
yIEF1dGhlbnRyY2F0b3IiLlCjhyWd1aWQiOiIwMTMyZDEwMC1iZjRlLTQyMDgtYTQwMy1hYjRmNWYxMmV
mZTUuLlCjhbHRlcm5hdG12ZURlc2NyaXB0aw9ucyI6eyJydS1SVSI6Itcf0YDQuNC80LXRgCBGSURPMiD
QsNGD0YLQtdC90YlQuNGE0LjQutCw0YLQvtGA0LAg0L7RgiBGSURPIEFsbG1hbmNlIiwiZnItRlIiOiJ
FeGVtcGx1IEZJRE8yIGF1dGhlbnRyY2F0b3IiZGUgRklETyBBBgxpYW5jZSI6InpoLUNOIjo15L6G6Ie
qRklETyBBBgxpYW5jZeeahOekuuS-i0ZJRE8y6Lqr5Lu96amX6K2J5ZmoIn0sInByb3RvY29sRmFtaWx
5Ijo1Zm1kbzIiLlCjZyZ2h1bWEiOjMsImF1dGhlbnRyY2F0b3JWZXJzaW9uIjo1LCJ1cHYiOi1t7Im1ham9
yIjoxLlCjtaW5vciI6MH1dLCJhdXRoZW50aWNhZGlvbkFsZ29yaXRobXMiOlSic2VjcDI1NnIxX2VjZHN
hX3NoYTI1N19yYXciLlCjyc2Fzc2FfcGtjc3YxNV9zaGEyNTZfcml3I10sInB1YmXpY0tleUFsZ0FuZEV
uY29kaW5ncyI6WyJjb3NlI10sImF0dGVzdGF0aw9uVHlwZXMiOlSiYmFzaWnfZnVsbCJdLCJ1c2VyVmV
yawZpY2F0aw9uRGV0YWlscyI6W1t7InVzZXJWZXJpZm1jYXRpb25NZXR0b2QiOiJub25lIn1dLft7InV
zZXJWZXJpZm1jYXRpb25NZXR0b2QiOiJwcmVzZW5jZV9pbmRlcm5hbCJ9XSxbeyJ1c2VyVmVyaWZpY2F
0aw9uTWV0aG9kIjoicGFzc2NvZGVfZXh0ZXJuYWwiLlCjYURlc2MiOnsiYmFzZSI6MTAsIm1pbkxlbmd
0aCI6NH19XSxbeyJ1c2VyVmVyaWZpY2F0aw9uTWV0aG9kIjoicGFzc2NvZGVfZXh0ZXJuYWwiLlCjYUR
lc2MiOnsiYmFzZSI6MTAsIm1pbkxlbmd0aCI6NH19LHsidXNlc1Zlcm1maWNhZGlvbk1ldGhvZCI6InB
yZXNlbmNlX2ludGVybmfSIn1dXSwia2V5UHJvdGVjdGlvbiI6WyJoYXJkd2FyZSI6InNlY3VyZV9lbGV
tZW50I10sIm1hdGNoZXJQcm90ZWNoaw9uIjpbIm9uX2NoaXAiXSwiY3J5cHRvU3RyZW5ndGgiOjEyOCw
iYXR0YWNobWVudEhpbmNlIiOlSic2VzZXh0ZXJuYWwiLlCj3aXJlZCI6IndpcmVsZXNzIiwibmZjI10sInRjRGJ
zcGxheSI6W10sImF0dGVzdGF0aw9uUm9vdENlcnRpZm1jYXRlcYI6WyJNSU1DUFRDQ0FlT2dBd0lCQWd
JSkFPdWV4d1UzT3kyd01Bb0dDQ3FHU000OUJBTUNNSHN4SURBZUJnTlZCQU1NRJfOaGJYQnNaU0JCZEh
SbGMzMUmhkR2x2Ym1CU2IyOTBNU1l3RkFZRFZRUUteEQTFHU1VSUE1FRnNiR2xoYm10bE1SRXdEd11EV1F
RTERBaFZRVVlnVkJkSExERVNNQkFHQTFVRUJ3d0pVR0ZzYn1CQmJlUnZNUXN3Q1FZRFZRUU1EQUpEUVR
FTE1Ba0dBmVVFQmhnNQ1ZWTxdIaGN0TVRRd05qRTRNVE16TXpNeVdoY050REV4TVRBek1UTXpNek15V2p
CN01TQXdIZ11EV1FRRERCZFRZVzF3YkdvZ1FYUjBaWE4wVWVhScGIyNgdVbTl2ZERFV01CUUdBMVVFQ2d
3TlJrbEVUeUJCyk4cFlXNWpaVEVSTUE4R0ExVUVDd3dJv1VGR0lGUlhSeXd4RwpBUUJnTlZCQWNNQ1Z
CaGJHOGdRV3gwYnpFTE1Ba0dBmVVFQ0F3Q1EwRXhDekFKQmdOVkJBWVRBbFZUTUZrd0V3WUhLb1pJemo
wQ0FRWU1Lb1pJemowREFRY0RRZ0FFSDhodjJEMeHYTU5L0JtcFE3U1plaEwwRk1HekZkMVFCZz12QV
wT1ozYwYpudVE5NFBsN2FNekgzM25VU0JyOGZlWURycU9CYjU4cHhHcUUhKUN1YLzZOUU1FNHdIUUV1EV1I
wT0JCWUVGUG9lQTNDTGh4RmJDMEl0N3pFNHc4aGs1RUovTUI4R0ExVWRJd1FZTUJhQUZQb0hBM0NMaHh
GYkMwSXQ3ekU0dzhoazVFSi9NQXdhHQTFVZEV3UUZNQU1CQWY4d0NnWU1Lb1pJemowRUF3SURTQUF3U1F
JaEFKMDZRU1h0OW1oSWJFS11LSWpzUGtyaVZkTE1ndGZzYkRTdTdFckpmenI0QWlCcw9ZQ1pmMct6STU
1YVFlQUhqSXpBOVhtNjNycnVBeEJa0XBz0XoyWE5sUT09I10sIm1jb24iOiJkYXRhOm1tYwdl13BuZzt
iYXNlNjQsaVZCT1J3MEtHZ29BQUFBTlNvaEVVZ0FBQUU4QUFBQXZDQV1BQUFDaXdkZmNBQUFBQVhOU1I
wSUFyczRjNlFBQUFBUM5RVTFQCQFDeGp3djhhZUUVBQUFBsMnFAfpjd0FBRHNNQUFBN0RBY2R2cUdRQUF
BYWhTVVJCvkdORddacjVieFJsr01mOUt6VEI4QU0vWUVoRTJXN3BRWmNXS0tCY2xTcEhBVGxFTeFSRTd
rTkVDQ0EzRmtXSzBDS0tTQ0Zj0tCY2dWQ0RXR05FU2RBWW1kd2dnZ0pCaVJpTWhGYy80d3k40Dg0enU
5TmrSbkdUZ1pKUDJuM25PKys40DkzM2Z2ZUJCeCtQcUN6SmtUVXZCYkxtcFVEV3ZCVELtcGNdu1p2WEx
DZfg5UjA1U2sx0WJiNWF0ZjU5OWZHKy91ckE1NDFxNDdhUDFMTFZh0VNJeVZOVWk4Sww4ZDVrR1RzaTM
wTkZ2N2Fp0W43UVpQTXdiZH1zMmVyVTJYTXFVZHK4K1pjYU5tR21tRTh5WE4zU1Vkm2ExOG5GMGZVbG9
2WiswQ1R6V3BkMlZqK2VPbTFiRX15NKR4NGk1cFVNR1d2ZW81MDZxMjI3ZHR1V0JJdWZmcjZvV3BwMEZ
QTkxob3cxNzUxTm0yMUx2UEgzc1Z0V2pmejY2TGZxbDh0WdDGuMw5WUZTWHTu3N1Yj1jZU9HY11rN01

OVWNHUGc4WnNiTWU5cmZRVWFhVi9KTVg5c3FkekRDU3ZwMGtaSG1UWmc5eDdiTEhjTW5UaGIXNmVKK21
WZlFxoHlhVVPRTkc2NGlYwiswL2txNnVPWkZPMFF0YXRkV0tmWG5SUTk5Qmo5MVI1T0lGbms1NGpOMG1
rVWlxbE8zWERXK01sKzk4bUtCNnRXN3JXcFpjUGmRMHpnNHRMc1lsVWM4NkU2ZUdEaklNdWJWcGN1c2V
hcmZnSVlHUmS2YnJoWlZyL0pjSHpvb0w3NTUwamVkJTEV4b3BXy0FwaTJaVXFodTdkTHZyVnNRVTgxemt
6T1BlZW1NUl12VnVRc1g3UGJpRFFZNUp2Wm9uZnRLKzFWWThIOXV0eDUzMGgwb2Iram1SWXFqNm91YVl
2RWVuVy9XbFlqcDhjd2JNBtY4MnRQd3FXMVI0dGovMlNIMTNJukpZbDRtb1p2WHBpU3FEcjdkWHRRSHh
hL1BLMy8rQldzSzFkVGdIdTZWOHRRSjNid0Zrd3BGclVPUTUwczFyM2xldm04elpjcte3K0JCYXc3Szh
sRUs1cXprWWVhcms5QThwN1AzR3pESytuZDNEUW93KzZVQzhTVk44Mm1ldjM4aw03TnRhWHRWMUNWcTZ
SZ3c0cGtzbWJkaTNidTJEZTDZmFCQnhjcWZ2cVbYVWpGUU5UUTIybGZkVZWVDY4c1RKs0Y1RG5TbVV
qZ2RxZzRtU1M5cG1zZkRKUjNHNlRvSDBpVzlhVjdMV0xIwVhLbGxURHQWTFRBDgtZSWFhbXAXuWpWdis
rdXlHVXhWZEowRE5WwFNtK2IxcVJ4cGw4NGRkZlgxTHAXty9kNj10c29kMHZzNWhHcmU5eHU4bytmcEx
SMWNHaE5URDZaNTdDOuTNV1hlZkpkt1o5NGJiOW9xZDFST25TN3FJVFR6SGltTXFPdmJPM2cwRGRWeWs
zV1FCaEJ6dEsZNV1LTmRPbmM4TzNhY1M2ZkRaRmdLYVhMc0VKcDVyZHZJsaUJxcDg5Y0pjcy9tN1R2czB
ya2pHZk40YjBrUG9abjNVSnVJT3JuWjIyeVAXzm12VXgrTzVnU3F1Y1YxbSt6U3VZT1ZocTdUV2JEaUx
WdmxqcGxMbG9wNkNMWFArMnF0dkdMSUwvMXZpbUlTZE1CZ3pTb0ZaeXU2VHFkK2p6eGdzUGFWOUJdCWV
lL05qWws2djZsSz1jd2lVYy9TVHRmMUHEcE0zYjU5Mnk3aDNUaHg1b3pLNj1ITHBZV3VBd2FxUzVjdjI
2cTdjZWI4ZWZWWFWSZVAzaUzV0HpqMwtuU3dawEhNbW5Da1kwT2dhbG83VVFmU0NNM3FRUXIySC9YR1A
3c3NYeDQ1Www5MUJ5ZUNlcDRtb1pvSCsxZkczeEQ0dFQ3eDhrd3lqOG53Yj1ldjI2VjBCNmQrN0g0ekt
2dWRBSDUzN0ZqcXl6T0hkSm5IRXV6bVhXL1dqE9idk5NYnY3bmh5d3NYMmFwc1d0QzgrNDhhTGVhcEU
3cDV3S1ppMEEyQVFSVjVudlI0RSt1SmMrYjYxa0FwcUluEJnbWQvNFY1UVAvbXQx0EhEQzdzUkhmdG1
ldTVsbWhwMHJUl0FMWDIzMMjXZDRCRm5EeDdWaTFjV1MydWZmMEliQjQ3cWV4eG1VajlRdXRZanVwZDN
0WUQ2YWJXQkJNcmgrYXBOYk9Lck5GMSt1Z0NhNHjPwEdmd01QUHRWaWF2aFUzWU1PQUFudVViL1IwN0w
weU9TZU9hZEU40EFwc1hGR2ZmMzB5bmhsSmdNNTFDVTZ2Tj1FemduCHZIQkZVewlWcmFlUGl3SjUzREY
1WlRabm9tRU5nODVrTlVkmM9KaTJXcHI0T21ta2ZONHg0ekhmaVZGYzhEdjh0enVoTnFPaWRpbEd2QTZ
ER3V1WndPNzhBQVFuNmNpRws2K3J3NVZjdmp2cU5EWVPb0lVd2FLU2hyeEF1WEExa0g0YVl1R2ZNWUR
jMTBXRjVUYTMxaFBKT2ZjVwhyVS9KbEl0aTZjNmVsU1lkQnBvNisrWWZqeDYxbEd0ZlJtNE1ENXJKMwo
zRm9HSG5qRFNCTmFyWVvNtUx5TXN6S3BiN3RYcG9IZlBzOGgzV3AXThpOZk5rNTRYeEMxd0RHVW1Ze1h
ZZWZoNnovY0t0Vm00RUJ4YTlWUUDeellyM0xyVU1SakhFS2trN3phRktZUUEyaEdRVTF6Kzgz1TkZXcFh
Ecmt6M3Z4MTBHcXhRnkj6ZU5ib0JrNW44azRuZwJSaCtRMWhXZnhURjBEMUV5V1VzNW52K2RnUXFLYXh
6dUNkRTBpc0hsMDJOUThhaDBtWHIXMkxhM20wZjl3aWs5K3dMTlRNWS84Nk1Qbzh5aTMxT2Z4bVQ2UFd
vcUc5K0RadWtZbmE1Nm1TwnQ1V1dTTeTVxVkExcndVeUpXWefSbnprawFpL2dIU0Q3UmtUewlob2dBQUF
BQkpSVTVFcmTKZ2dnPT0iLCJzdXBwb3J0ZWRFeHRlbnNpb25zIjpbeyJpZCI6ImhtYWmtc2VjcmV0Iiw
iZmFpbF9pZl91bmtub3duIjpmYXxzZX0seyJpZCI6ImNyZWRCm90ZWNOIiwiaG1hYy1zZWNYZXQiXSwiYWFnZWlkIjo
uIjpmYXxzZX1dLCJhdXRoZW50aWNhdG9yR2V0SW5mbyI6eyJ2ZXJzaW9ucyI6WyJVMkZfVjIiLCJGSUR
PXZjFmCjdlLCJleHRlbnNpb25zIjpbImNyZWRCm90ZWNOIiwiaG1hYy1zZWNYZXQiXSwiYWFnZWlkIjo
iMDEzMmQxMTBiZjZjRlNDIwOGE0MDNhYjRmNWYxMmVmZTUuLCJvcHRpb25zIjpb7InBsYXQiOiJmYXxzZSI
sInJrIjoidHJ1ZSIzImNsaWVudFBpbIi6InRydWUiLCJ1cCI6InRydWUiLCJ1diI6InRydWUiLCJ1dIR
va2VuIjoiZmFsc2UiLCJjb25maWciOiJmYXxzZSJ9LCJtYXhNc2dTdTaXp1IjoxMjAwLCJwaW5VdkF1dGh
Qcm90b2NvbHMlOlssXSwibWf4Q3JlZGVudG1hbENvdW50SW5MaXN0IjoxNiwbWf4Q3JlZGVudG1hbEl
kTGvUz3RoIjoxMjg5InRyYW5zcG9ydHMiOlssidXNiIiwibmZjIl0sImFsZ29yaXRobXMiOl0t7InR5cGU
iOiJwdWJsawMta2V5IiwiaWxnbmFpbGUiOiJmYXxzZSJ9LCJtYXhNc2dTdTaXp1IjoxMjAwLCJwaW5VdkF1dGh
tYXhBdXRoZW50aWNhdG9yQ29uZm1nTGvUz3RoIjoxMDI0LCJkZWZhdWx0Q3JlZFBYb3RlY3QiojIjImZ
pcm13YXJlVmVyc2lubiI6NX19LCJzdGF0dXNSZXBvcnRzIjpbeyJzdGF0dXMiOiJGSURPX0NFU1RJRk1
FRClSImVmZmVjdG12ZURhdGUiOiIyMDE5LTAxLTA0In0seyJzdGF0dXMiOiJGSURPX0NFU1RJRk1FRF9
MMSIsImVmZmVjdG12ZURhdGUiOiIyMDIwLTExLTExIiwiaWxnbmFpbGUiOiJmYXxzZSJ9LCJtYXhNc2dTdTaXp1IjoxMjAwLCJwaW5VdkF1dGh
GSURPIEFsbG1hbmNlIFNhbXBsZSBGSURPMiBBdXRoZW50aWNhdG9yIiwiaWxnbmFpbGUiOiJmYXxzZSJ9LCJtYXhNc2dTdTaXp1IjoxMjAwLCJwaW5VdkF1dGh

```
i0iJGSURPMjEwMDayMDE1MTIyMTAwMSIsImNlcnRpZmljYXRpb25Qb2xpY3lWZXJzaW9uIjoMS4wLjE  
iLCJjZXJ0awZpY2F0aw9uUmVxdWlyZW11bnRzVmVyc2lvaWwI6IjEuMC4xIn1dLCJ0aw11T2ZMYXN0U3R  
hdHVzQ2hhbmdlIjoIMjAxOS0wMS0wNCJ9XX0
```

and finally we have to append another period (".") followed by the base64url-encoded signature.

EXAMPLE: JWT

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCIsIng1YyI6WyJNSU1DWlRDQ0FndWdBd0lCQWdJQkFUQUtC  
Z2dxaGtqT1BRUURBakNCb3pFbk1DVUdBMVVFQXd3ZVJwaEJUVkKJNU1NCT1JGTXpJR1JGVTRFRZ1NVNVVS  
VkpOU1VSS1FWUKZNU013SUFZSkthVWl0dmNOQVFRQkZoTmx1R0Z0Y0d4bFFHVjRZVzF3YkdVdVkyOXRN  
U1F3RwdZRFZRUUteEQXRGZUdGdGNHeGxJRTlTUnpFUU1BNEdBWVVFQ3d3SFJYbGhiWEJzWlRfTE1Ba0dB  
MVVFQmhmNQ1ZWtXhDekFKQmdOVkJBZ01BazFaTVJjd0VBWURWUVFIREFsWF1XdGxabWxsYkdRd0hoY05N  
akV3TkRfNU1URXpOVEEzV2hjTk16RXdOREUzTVRFek5UQTNXakNCcFRFcE1DY0dBWVVFQXd3Z1JWaEJU  
VkJNU1NCT1JGTXpJRk5KUjA1SlRyY2dRMFZTVkVsR1NVTKJWRVV4SwBZ0Jna3Foa2lH0XcwQkNRRVdF  
MlY0WVcxZD1H0VfFawGhoYlhcC1pTNWpimjB4RkRBU0JnTlZCQW9NQzBWNFlXMXdiR1VnVDFKSE1SQXdE  
Z1lEVlFRTERBZEZlR0Z0Y0d4bE1Rc3dDUVlEVlFRR0V3SlZVekVMTUFR0ExVUVDQXdDVfZreEVqQVFC  
Z05WQkFjTUNWZGhm1ZtYVdwc1pEQlPnQk1HQnlxR1NNND1BZ0VHQ0N0R1NNND1Bd0VIQTBJQUJOUUpz  
NndUcWl4YytTK1ZEQWFqRmxQTmF0MTBLRVdKRTVqY1dPdm02cXBPOVNEQUFNWnZiNEhIcnZk1A1WVJw  
SHJTBfVQZHZLK3VFUWJkV2czMVA5dWpMREffTUFR0ExVWRf1FDTUFBd0hRWURWUjBPQkJZRUZMcXNh  
cGNyVjRab1ZlQW5ScFBad1FlN1l5MjBNQW9HQ0N0R1NNND1CQU1DQTBnQU1FVUNJUUM2N3ph0EVJdXlS  
aUtnTKRYSVAxczFhTHIzanpIOVdWwGZiEDRiSit6Q3NnSWdHL3RWQnV0T0pVvSt2dm9ISW8vb3RBVUFj  
SDViTkhQM3VJemlEUytQVfVjPSIsIk1JSUVIEkNDQWdlZ0F3SUJBZ0lCQWpBTkKna3Foa2lH0XcwQkFR  
c0ZBRENCbXpFZk1CMEdBMVVFQXd3V1JwaEJUVkKJNU1NCT1JGTXpJR1JGVTRFRZ1Vr0VBWREVPtUNBR0NT  
cUdTSWIZRFFFSkFSWVRaWghoYlhcC1pVQmx1R0Z0Y0d4bExtTnZiEVVtUJJR0ExVUVDZ3dMUlhoaGJY  
QnNaU0JQVWtjeEVEQU9CZ05WQkFzTUWVjRZVzF3YkdVeEN6QU9CZ05WQkFZVEFsV1RNUXN3Q1FZRFZR  
UU1EQUpOV1RFU01CQUdBMVVFQnd3SlYyRnJaV1pwwld4a01CNFhEVE14TURReE9URXhNe1V3TjFvWERU  
UTRNRgt3TKRfEE16VXdOMW93Z2FNeEp6QWxCZ05WQkFNTUhrV1lRVTRFRVZ3dMUlRNeUJVVU1ZOVU1F  
bE9WRVZTVFVWRVNV1VSVEVPtUNBR0NTcUdTSWIZRFFFSkFSWVRaWghoYlhcC1pVQmx1R0Z0Y0d4bExt  
TnZiEVVtUJJR0ExVUVDZ3dMUlhoaGJYQnNaU0JQVWtjeEVEQU9CZ05WQkFzTUWVjRZVzF3YkdVeEN6  
QU9CZ05WQkFZVEFsV1RNUXN3Q1FZRFZRUU1EQUpOV1RFU01CQUdBMVVFQnd3SlYyRnJaV1pwwld4a01G  
a3dFd1lIS29aSXpqMENBUVlJS29aSXpqMERBUWNEUWdBZ05HdW1CY1luR1FuVWpQVWJZmM3MGhzaGdi  
aUkxWnRwd1E1bjZ4UkxBL1dxMFBTQ2ZMbDVxUSTyN2RsY0sxZDNyM3ZMYSt2bTZHNnZlSEdDUEVlVXpx  
TXZNQzB3REFZRFZSMFRQV3QXdfQi96QWRCZ05WSFE0RUZnUUV0azZGNFJKbkdhVWZlKzAvY2Jad2Zy  
WmQ3WlV3RFFZSkthVWl0dmNOQVFFTEJRQRnZ0lCQUUucDFmbTBGS2xXbVv0VHBsTHVZZZdtcHM0eFAV  
Q0910GRuYjM4dTFuTURWdU9UNCtDWMfPTT1BR3ozMTNHRDIyaGpMR3JtUHVZbjg2d0dPS0kzSE9yRXBz  
R2RnbWZ5N3RUBUtYL2VNL2VTM0ZFRFhabkU4M1BuNW9GSX1CVC9mOHNHdVh5T3NGWnFXQnZWZEJJJSURs  
ZENwRDRteE1RWlpPWNrUcmx2M1d2Q1fNqY9kC2lJt3h1M1FLWHZXSGk2UWivUmh1Yw1wM3JQbXdnZis0  
SnBuSk8rSk1QcUFhVTFjQUg4SFZzZnJMQU1vS3MxNDhqMitjdmJwYVdte1Q1cklvSC9lelZyUGFHL01P  
aUlnTc5dy9lZnV2U2k1QVg4SitrRG9MU0VmM2Q1d09na0pZQXFVcWNSefHURUV0S016RE02aHphQ1FG  
aUFxd1RuOUlsVldnbnRRYw1TWHZIK3R4YVRGOWlfbEh4VWY1SU5ZRlZjaUNwenRTcnlkZUh2L09DTlJm  
Ny9MVnJpY01TbG84UmgrTzN5UDlWkZj1TmYzWdhzUUpOdHVMclFOYXFxMTh3aVhsaVRmdWZTbjAyL2cr  
bWtoSVVpTktmVE9KcHZDaktlQ25DRmN4UVUyL1hUM0toM0c4Z0RKd3NPNkVWUmpNVUp0NEFZS3p1L2hF  
VUN3RjU1SUYybTNqSE1vQ3U4alZmajI0Q2VFWDVkbmZ2U3IrU1Z2TjVRQjB1WjA1TTRYbXlaWHLxQm0w  
eksZlIraUuWl1pSW51d0x0DN1grVzgyelhsbk1rcGxJM1ErSnhkn2pmUTE1U1lORTJLNN2UklUMDF3  
MFA5WnF5REY3a25HS3BSbHA3T3F4ZDM3YkQvV1ViV3BRN2dJQWZzSk5INutCTG93SEpGRmpXI119.eyJ  
sZWdhbEh1YWRlciI6Ij1ldHJpZXZhbCBhbmQgdXNlIG9mIHROaXMGkxPQiBpbmRyY2F0ZXMGYWNjZXB
```

0YW5jZSBvZiB0aGUYXBWcm9wcm1hdGUGYWdyZWVtZW50IGxvY2F0ZWQgYXQgaHR0cHM6Ly9maWRvYXVw
saWFuY2Uub3JnL21ldGFkYXRhL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlbnRyaWVzIjpbeyJhYWlkIjoimTIzNCM1Njc4IiwibWV0YWRhdGFtdGF
0ZW1lbnQiOnsibGVnYXwIZWfkZXIIoiJodHRwcovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV
0YWRhdGEtc3RhdGVtZW50LWxlZ2FsLWlhYWRlci8iLCJkZXNjcm1wdGlvbiI6IkZJRE8gQWxsawFuY2U
gu2FtcGx1IFVBRiBBdXRozW50aWnhdG9yIiwiYWFPZCI6IjEyMzQjNTY3OClSIImFsdGVybmf0aXZlRGV
zY3JpcHRpb25zIjp7InJlLVJVIjojRgNC40LzQtdGAIFVBRiDQsNGD0YLQtdC90YlQuNGE0LjQutC
w0YLQvtGA0LAG0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJFeGVtcGx1IFVBRiBhdXRozW50aWn
hdG9yIGRlIEZJRE8gQWxsawFuY2UifSwiYXV0aGVudGljYXRvc1ZlcnNpb24iOjIsInByb3RvY29sRmF
taWx5IjojdWFMiIiwic2NoZW1hIjozLCJ1cHYiOlt7Im1ham9yIjoxL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlbnRyaWVzIjpbeyJhYWlkIjoimTIzNCM1Njc4IiwibWV0YWRhdGFtdGF
0ZW1lbnQiOnsibGVnYXwIZWfkZXIIoiJodHRwcovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV
0YWRhdGEtc3RhdGVtZW50LWxlZ2FsLWlhYWRlci8iLCJkZXNjcm1wdGlvbiI6IkZJRE8gQWxsawFuY2U
gu2FtcGx1IFVBRiBBdXRozW50aWnhdG9yIiwiYWFPZCI6IjEyMzQjNTY3OClSIImFsdGVybmf0aXZlRGV
zY3JpcHRpb25zIjp7InJlLVJVIjojRgNC40LzQtdGAIFVBRiDQsNGD0YLQtdC90YlQuNGE0LjQutC
w0YLQvtGA0LAG0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJFeGVtcGx1IFVBRiBhdXRozW50aWn
hdG9yIGRlIEZJRE8gQWxsawFuY2UifSwiYXV0aGVudGljYXRvc1ZlcnNpb24iOjIsInByb3RvY29sRmF
taWx5IjojdWFMiIiwic2NoZW1hIjozLCJ1cHYiOlt7Im1ham9yIjoxL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlbnRyaWVzIjpbeyJhYWlkIjoimTIzNCM1Njc4IiwibWV0YWRhdGFtdGF
0ZW1lbnQiOnsibGVnYXwIZWfkZXIIoiJodHRwcovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV
0YWRhdGEtc3RhdGVtZW50LWxlZ2FsLWlhYWRlci8iLCJkZXNjcm1wdGlvbiI6IkZJRE8gQWxsawFuY2U
gu2FtcGx1IFVBRiBBdXRozW50aWnhdG9yIiwiYWFPZCI6IjEyMzQjNTY3OClSIImFsdGVybmf0aXZlRGV
zY3JpcHRpb25zIjp7InJlLVJVIjojRgNC40LzQtdGAIFVBRiDQsNGD0YLQtdC90YlQuNGE0LjQutC
w0YLQvtGA0LAG0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJFeGVtcGx1IFVBRiBhdXRozW50aWn
hdG9yIGRlIEZJRE8gQWxsawFuY2UifSwiYXV0aGVudGljYXRvc1ZlcnNpb24iOjIsInByb3RvY29sRmF
taWx5IjojdWFMiIiwic2NoZW1hIjozLCJ1cHYiOlt7Im1ham9yIjoxL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlbnRyaWVzIjpbeyJhYWlkIjoimTIzNCM1Njc4IiwibWV0YWRhdGFtdGF
0ZW1lbnQiOnsibGVnYXwIZWfkZXIIoiJodHRwcovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV
0YWRhdGEtc3RhdGVtZW50LWxlZ2FsLWlhYWRlci8iLCJkZXNjcm1wdGlvbiI6IkZJRE8gQWxsawFuY2U
gu2FtcGx1IFVBRiBBdXRozW50aWnhdG9yIiwiYWFPZCI6IjEyMzQjNTY3OClSIImFsdGVybmf0aXZlRGV
zY3JpcHRpb25zIjp7InJlLVJVIjojRgNC40LzQtdGAIFVBRiDQsNGD0YLQtdC90YlQuNGE0LjQutC
w0YLQvtGA0LAG0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJFeGVtcGx1IFVBRiBhdXRozW50aWn
hdG9yIGRlIEZJRE8gQWxsawFuY2UifSwiYXV0aGVudGljYXRvc1ZlcnNpb24iOjIsInByb3RvY29sRmF
taWx5IjojdWFMiIiwic2NoZW1hIjozLCJ1cHYiOlt7Im1ham9yIjoxL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlbnRyaWVzIjpbeyJhYWlkIjoimTIzNCM1Njc4IiwibWV0YWRhdGFtdGF
0ZW1lbnQiOnsibGVnYXwIZWfkZXIIoiJodHRwcovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV
0YWRhdGEtc3RhdGVtZW50LWxlZ2FsLWlhYWRlci8iLCJkZXNjcm1wdGlvbiI6IkZJRE8gQWxsawFuY2U
gu2FtcGx1IFVBRiBBdXRozW50aWnhdG9yIiwiYWFPZCI6IjEyMzQjNTY3OClSIImFsdGVybmf0aXZlRGV
zY3JpcHRpb25zIjp7InJlLVJVIjojRgNC40LzQtdGAIFVBRiDQsNGD0YLQtdC90YlQuNGE0LjQutC
w0YLQvtGA0LAG0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJFeGVtcGx1IFVBRiBhdXRozW50aWn
hdG9yIGRlIEZJRE8gQWxsawFuY2UifSwiYXV0aGVudGljYXRvc1ZlcnNpb24iOjIsInByb3RvY29sRmF
taWx5IjojdWFMiIiwic2NoZW1hIjozLCJ1cHYiOlt7Im1ham9yIjoxL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlbnRyaWVzIjpbeyJhYWlkIjoimTIzNCM1Njc4IiwibWV0YWRhdGFtdGF
0ZW1lbnQiOnsibGVnYXwIZWfkZXIIoiJodHRwcovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV
0YWRhdGEtc3RhdGVtZW50LWxlZ2FsLWlhYWRlci8iLCJkZXNjcm1wdGlvbiI6IkZJRE8gQWxsawFuY2U
gu2FtcGx1IFVBRiBBdXRozW50aWnhdG9yIiwiYWFPZCI6IjEyMzQjNTY3OClSIImFsdGVybmf0aXZlRGV
zY3JpcHRpb25zIjp7InJlLVJVIjojRgNC40LzQtdGAIFVBRiDQsNGD0YLQtdC90YlQuNGE0LjQutC
w0YLQvtGA0LAG0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJFeGVtcGx1IFVBRiBhdXRozW50aWn
hdG9yIGRlIEZJRE8gQWxsawFuY2UifSwiYXV0aGVudGljYXRvc1ZlcnNpb24iOjIsInByb3RvY29sRmF
taWx5IjojdWFMiIiwic2NoZW1hIjozLCJ1cHYiOlt7Im1ham9yIjoxL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlbnRyaWVzIjpbeyJhYWlkIjoimTIzNCM1Njc4IiwibWV0YWRhdGFtdGF
0ZW1lbnQiOnsibGVnYXwIZWfkZXIIoiJodHRwcovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV
0YWRhdGEtc3RhdGVtZW50LWxlZ2FsLWlhYWRlci8iLCJkZXNjcm1wdGlvbiI6IkZJRE8gQWxsawFuY2U
gu2FtcGx1IFVBRiBBdXRozW50aWnhdG9yIiwiYWFPZCI6IjEyMzQjNTY3OClSIImFsdGVybmf0aXZlRGV
zY3JpcHRpb25zIjp7InJlLVJVIjojRgNC40LzQtdGAIFVBRiDQsNGD0YLQtdC90YlQuNGE0LjQutC
w0YLQvtGA0LAG0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJFeGVtcGx1IFVBRiBhdXRozW50aWn
hdG9yIGRlIEZJRE8gQWxsawFuY2UifSwiYXV0aGVudGljYXRvc1ZlcnNpb24iOjIsInByb3RvY29sRmF
taWx5IjojdWFMiIiwic2NoZW1hIjozLCJ1cHYiOlt7Im1ham9yIjoxL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlbnRyaWVzIjpbeyJhYWlkIjoimTIzNCM1Njc4IiwibWV0YWRhdGFtdGF
0ZW1lbnQiOnsibGVnYXwIZWfkZXIIoiJodHRwcovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV
0YWRhdGEtc3RhdGVtZW50LWxlZ2FsLWlhYWRlci8iLCJkZXNjcm1wdGlvbiI6IkZJRE8gQWxsawFuY2U
gu2FtcGx1IFVBRiBBdXRozW50aWnhdG9yIiwiYWFPZCI6IjEyMzQjNTY3OClSIImFsdGVybmf0aXZlRGV
zY3JpcHRpb25zIjp7InJlLVJVIjojRgNC40LzQtdGAIFVBRiDQsNGD0YLQtdC90YlQuNGE0LjQutC
w0YLQvtGA0LAG0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJFeGVtcGx1IFVBRiBhdXRozW50aWn
hdG9yIGRlIEZJRE8gQWxsawFuY2UifSwiYXV0aGVudGljYXRvc1ZlcnNpb24iOjIsInByb3RvY29sRmF
taWx5IjojdWFMiIiwic2NoZW1hIjozLCJ1cHYiOlt7Im1ham9yIjoxL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0

wWmNQYyswemc0dExyWwXVYzg2RTZlR0RqSU11YlZwY3VzZWfYzmdJWUdSazZicmhaVnIvSmNIem9vTDc
1NTBqZWRRMXhvcFdjQXBpMlpVcWh1N0pMdnJwc1FVODF6a3pPUGVlbU1SWXZWdVFzWdDQYm1EUvk1SnZ
ab25mdEsRmVZZOEG5dXR4NTMwaDBvYitqbVJZcwo2b3VhWXFZW5XL1dsWWpwOGN3Yk1tNjgydFB3cVc
xUjR0ai8yU0gxM0lSS1lsNG1vWnZYcGlTcURyN2RYdFFIEGEvUESzLytCV3NLMWRUZ0h1NlY4dFFKM2J
3Rmt3cEzyVU9RNTBzMXIzbGV2bTh6WmNxMTcrQkJhdzdLOGxFSzVxemtZZWfYaz1BOHA3UDNHekRLK25
kM0RRb3crNlVDOFNWTjgyaXV2MzhpbTdOdGFYdFYxQ1ZxNlJndzRwa3NtYmRPM2J1MkRlN1lmYUJCeGN
xZnZxUHJVakZRTlRRMjJsZmRVVlZUNjhyVEpLRjVEblNtVWpnZHFnNG1TUz1wbXNmREpSM0c2VG9IMG1
XOWFWN0xXTEhZWEtsbFREDDBMVEF0a1lJYWfTcDFRa1Z2Kyt1eUdVeFZkSjBET1ZYU20rYjFxFxUnhwbDg
0ZGRmWDFMcDFPL2Q20XRzb2QwdnM1aEdyZT14dThvK2ZWtFiXy0doT1RENlo1N0M5S01XWGVmSmRPWjk
0YmI5b3FkMVJPblM3cUlUVHpIaw1NcWl2Yk8zZzBEZFZ5azNXUUJoQnp0SzM1WU0ZE9uYzhPM2FjUzZ
mRFpGZ0thWExzRUUpwNXJkcmxpQnFwOD1jSmNzL203VHZzMHJrakdmTjRiMGtQb1puM1VKdUlPcm5aMjJ
5UDFmbXZVeCtPNwdTcWviVjFtK3pTdVl0VmhxN1RXYkRpTFZ2bGpwbExsb3A2Q0xYUCsycXR2R0xJTC8
xdm1tSVNkTUJne1NvRlp5dTZUcWQranp4Z3NQYVY5QkNxZWUvTnpZazZ2NmXLOWN3aVVjL1NUdGYxSER
wTTNiNTkyeTdoM1RoeDVveks2OUhMcF1XdUF3YXFTNWN2MjZxN2NlYjh1ZlZZYVJlUDNpRlU4emoxa25
Td1pYSE1tbkNqWTBPZ2FsbzdVUWZTQ00zcVFRcjJIL1hGUDdzc1h4NDVZbDkxQn1lQ2VwNG1vWm9IKzF
mRzN4RDR0VDd40Gt3eWo4bndiOWV2MjZWMEI2ZCs3SDR6S3Z1ZEFINTM3RmpxeXpPSGRKbkHfDxptWHE
vV2p4T2J2Tk1idjduaHl3c1gyYVZzV3RDOCs00GFMZWfWRTdWNXdLWmkwQTJBuVJWNW52UjRfK3VKYyt
iNjFrQXBxSW54QmdtZC80VjVRUC9tdDE4SERDN3NSSGZ0bWV1NWxtaFYwcm4vQUxYmJMyYnFkNEJGbkR
4N1ZpMWNXUzJ1ZmYwSWJCNDdxZXh4bVVqOVF1dFlqdXBkM3RZRDZhYldCQk1yaCthcE5iT0tyTkYxK3V
nQ2E0cm1YR2Z3TVBQdFZpYXZoVTNztU9BQW51VWIVUjA3TDB5T1N1T2FkRTg4QXBzWEZHmYzMHlUaGx
KZ001MUNVnNz00UV6Z25wdkhCR1V5aVzyYwVQaXdKNTNERjVaVfpub21FTmc4NwtOVWQyb0ppMldwcjR
PbW1rZk40eDR6SGZpVkJzjOER2OE56dWhOcU9pZGlsR3ZBNkRHdWVad083OEFBUW42Y2lFazYrcnc1VmN
2anZxTkRZUE9vSVV3YUtTaHJ4QXVYTgxRSDRhWVXVHk1ZRGmXMFdGNVRhMzFoUEpPZmNVaHJVL0psSU5
pNmM2ZWxSWWRCcG82KytZZmp4NjFsR05mUm00TUQ1ckoxajNgB0dIbmpEU0JOYXJZVWdNTH1Nc3pLcGI
3dFhwb0hmUHM4aDNXcDFMek5mTms1NFh4QzF3REdVbVl6WFl1Zmg2ei9jS3RwbTRFQnhhOVZRR0R6WXI
zTHJVTVJqSEVL2s3emFGS1lRQTJoR1FVMXorODVORldwERYa3ozdngxMEdxFe2Qnp1TmJvQms1bjh
rNG5lYlJoK2sxaFdmeFRGMEQxRXlXVXM1bnYrZGdRcUtheHp1Q2RFMG1zSGwwMk5ROGFoMG1YcjEyTGE
zbTBM0XdpazkrD0x0VE1ZLzg2TVBvOHlpMzFPZnhtVDZQV29xRzkrRFp1a1luYTU2bVNadDVXV1N5NXF
WQTFyd1V5SnFYQWxuemptYwkvZ0hTRDdSa1R5aWhvZ0FBQUFCS1JVNUVya0pnZ2c9PSJ9LCJzdGF0dXN
SZXBvcnRzIjpbeyJzdGF0dXMiOiJGSURPX0NFU1RJRklFRClSImVmZmVjdGl2ZURhdGUiOiIyMDE0LTA
xLTA0In1dLCJ0aw1lT2ZMYXN0U3RhdHVzQ2hhbmdlIjoimJAxNC0wMS0wNcJ9LHsiYWFndWlkIjoimDE
zMmQxMTAtYmY0ZS00MjA4LWE0MDMtYWI0ZjVmMTJlZmU1IiwibWV0YWRhdGFtdGF0ZW1lbnQiOmsibGV
nYWxIZWfKZXIiOiJodHRwcZovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbwV0YWRhdGEtc3RhdGV
tZW50LWxlZ2FsLWhlYWRLci8iLCJkZXNjcmlwdGlvbiI6IkZJRE8gQWxsawFuY2UgU2FtcGxlIEZJRE8
yIEF1dGhlbnRyY2F0b3IiLCJhYWdlawQI0iIwMTMyZDExMC1iZjRlLTQyMDgtYTQwMy1hYjRmNWYxMmV
mZTUuIiLCJhbHRlcm5hdG12ZURlc2NyaXB0aw9ucyI6eyJydS1SVSI6Itcf0YDQuNC80LXRgCBGSURPMiD
QsNGD0YLQtdC90YLQuNGE0LjQutCw0YLQvtGA0LAG0L7RgiBGSURPIEFsbG1hbmNlIiwiznItRlIiOiJ
FeGvtcGxlIEZJRE8yIGF1dGhlbnRyY2F0b3IgzUGUgRklETyBBbGxpYW5jZSIsInpoLUN0Ijoim5L6G6Ie
qRklETyBBbGxpYW5jZeeahOekuuS-i0ZJRE8y6Lqr5Lu96amX6K2J5ZmoIn0sInByb3RvY29sRmFtaWx
5Ijoim1kbziIiLCJzY2h1bWEiOjMsImF1dGhlbnRyY2F0b3JWZXJzaW9uIjoim1LCJ1cHYiOlt7Im1ham9
yIjoxLCJtaW5vciI6MH1dLCJhdXR0ZW50awNhdGlvbkfS29yaXRobXMiOlsic2VjcDI1NnIxX2VjZHN
hX3NoYTI1Nl9yYXciLCJyc2Fzc2FfcGtjc3YxNV9zaGEyNTZfcmf3I10sInB1YmXpY0tleUFsZ0FuZEV
uY29kaw5ncyI6WyJjb3NlI10sImF0dGVzdGF0aw9uVHlwZXMiOlsiYmFzaWNfZnVsbcjJdLCJ1c2VyVmV
yaWZpY2F0aw9uRGV0YWLscyI6W1t7InVzZXJWZXJpZm1jYXRpb25NZXR0b2QiOiJub25lIn1dLft7InV
zZXJWZXJpZm1jYXRpb25NZXR0b2QiOiJwcmVzZW5jZV9pbmRlcm5hbCJ9XSxbeyJ1c2VyVmVyaWZpY2F
0aw9uTWV0aG9kIjoicGFzc2NvZGVfZXh0ZXJyYWRlcjYURlc2MiOmsiYmFzZSI6MTAsIm1pbkxlbmd

0aCI6NH19XSxbeyJ1c2VyVmVyaWZpY2F0aW9uTWV0aG9kIjoicGFzc2NvZGVfZXh0ZXJyYWwiLCJjYUR
lc2MiOnsiYmFzZSI6MTAsIm1pbkxlbmd0aCI6NH19LHsidXNlc1Zlcm1maWnhdGlvbk1ldGhvZCI6InB
yZXNlbmNlX2ludGVybmfSIn1dXSwia2V5UHJvdGVjdGlvbiI6WyJoYXJkd2FyZSIsInNlY3VyZV9lbGV
tZW50I10sIm1hdGNoZXJQcm90ZWN0aW9uIjpbIm9uX2NoaXAiXSwiY3J5cHRvU3RyZW5ndGgiOjEyoCw
iYXR0YWNobWVudEhpbnQiOlsiZXh0ZXJyYWwiLCJ3aXJlZCI6IndpcmVsZXNzIiwibmZjI10sInRjRG1
zcGxheSI6W10sImF0dGVzdGF0aW9uUm9vdENlcnRpZm1jYXRlcYI6WyJNSU1DUFRDQ0F1T2dBd0lCQWd
JSkFPdWV4d1UzT3kyd01Bb0dDQ3FHU0000UJBTUNNSHN4SURBZUJnTlZCQU1NRjFOaGJYQnNaU0JCZEh
SbGMzUmhkR2x2Ym1CU2IyOTBNUL13RkFZRFZRUUteQTFHU1VSUElFRnNiR2xoYm10bE1SRXdEd1lEVlF
RTERBafZRVVlNvkZkSExERVNNQkFHQTFVRUJ3d0pVR0ZzYn1CQmJIUnZNUNX3Q1FZRFZRUU1EUpeUVR
FTE1Ba0dBmVVFQmhnNQ1ZWtXDIaGNOTVRRD05qRTRNVE16TXpNeVdoY05OREV4TVRBek1UTXpNek15V2p
CN01TQXdIZ1lEVlFRERERCZFRZvZf3YkdVZ1FYUjBaWE4wVhScGIyNGdVbTl2ZERFV01CUUdBmVVFQ2d
3TlJrbEVUeUJCYkd4cFlXNwpaEVSTUE4R0EXVUVDd3dJVlVGR0lGUlhSeXd4RwPBUUJnTlZCQWNNQ1Z
CaGJHOGdRV3gwYnpFTE1Ba0dBmVVFQ0F3Q1EwRXhDekFKQmdOVkJBWVRBbFZUTUZrd0V3WUHLb1pJemo
wQ0FRWU1Lb1pJemowREFRY0RRZ0FFSDhodjJEMehYYTU5L0JtcFE3U1plaEwwRk1HekZkMVFCZz12QVV
wT1ozYWpudVE5NFBsN2FNekgzM25VU0JyOGZIWURycU9CYjU4cHhHcUHKUn1YLzZOUU1FNHdIUUVlEVlI
wT0JCWUVGUG9lQTNDTGh4RmJDMEl0N3pFNHc4aGs1RUovTUI4R0EXVWRJd1FZTUJhQUZQb0hBM0NMaHh
GYkMwSXQ3ekU0dzhoazVFSi9NQXDhQTFVZEV3UUZnQU1CQWY4d0NnWU1Lb1pJemowRUF3SURTQUF3U1F
JaEFKMDZRU1h00WloSWJFS1lLSwpzUGtyaVZkTElndGZzYkRTdTdFckpmenI0QWlCcw9ZQ1pmMct6STU
1YVF1QUhqSXpBOVhtNjNycnVBeEJa0XBz0XoyWE5sUT09I10sIm1jb24iOiJkYXRhOm1tYWdl1L3BuZzt
iYXNlNjQsaVZCT1J3METHz29BQUFBTlNVaEVVZ0FBQUU4QUFBQXZDQVlBQUFDaXdKZmNBQUFBQVhOU1I
wSUFyczRjNlFBQUFBUM5RVTFcQUFDeGp3djhzUUVVBQUFBsmNFaFpjd0FBRHNNQUFBN0RBY2R2cUdrQUF
BYWhTVVJCvkd0RDdaczVieFJsR01mOut6VEI4QU0vWUVoRTJXN3BRWmNXS0tCY2xtCEhBVGxfTEFSRTd
rTkVDQ0EzRmtXSzBDS0tTQ0Zjc0tCY2dWQ0RXR05FU2RBWw1kd2dnZ0pCaVJpTWhGYy80d3k40DG0enU
5TmRsbkdUZlPkUDJuM25PKys40DkzM2Z2ZUJCeCtQcUN6SmtUVXZCYkxtcFVEV3ZCVELtcGNDU1p2WEx
DZFG5UjA1U2sx0WJiNWF0ZjU50WZHky9lcke1NDFxNDdhUDFMTFZhOVNJeVZ0VWk4SWk4ZDVRr1RzaTM
wTkZ2N2Fp0W43UVpQTXdiZHlzMmVyVTJYTXFVZHk4K1pjYU5tR21tRTh5WE4zU1Vkm2ExOG5GMGZVbG9
2WiswQ1R6V3BkMlZqK2VPbTFiRX15Nkr4NGk1cFVNR1d2ZW81MDZxMjI3ZHR1V0JJdWZmcjZvV3BWMEZ
QTkxob3cxNzUxTm0yMUx2UEgzclZ0V2pmejY2TGZxbDh0WDdGUmw5WUZZTWHNtU3N1Yj1jZU9HY11rN01
OVWNHUGc4WnNiTWU5cmZRVWFhVi9KTVg5c3FkekRDU3ZWmGtaSG1UWmc5eDdiTEhjTW5UaGIXnmVKK21
WZlFxoHlhVVPRTkc2NGlYwiswL2txNnVPWkZPMFF0YXRkv0tmWG5SUTk5Qmo5MVI1T0lGbms1NGpOMG1
rVWlxbE8zWERXK01sKzk4bUtCNnRXN3JXcFpjUGMRMHpnNHRMc1lsVWM4NkU2ZUdEaklNdWJWcGN1c2V
hcmZnSVlHums2YnJoWlZyL0pjSHpvb0w3NTUwamVKEV4b3BXy0FwaTJaVXFodTdkTHZyVnNRVTgxemt
6T1BlZW1NU112VnVRc1g3UGJpRFFZNUp2Wm9uZnRLKzFWWThIOXV0eDUzMGgb2Iram1SWXFqNm91YVl
2RWVuVy9XbFlqcDhjd2JNbTY4MnRQd3FXMVI0dGovMlNIMTNJUkpZbDRtb1p2WHBpU3FEcjdkWHRRSHh
hL1BLMy8rQldzSzFkVGdIdTZW0HRRSJNid0Zrd3BGclVPUTUwcZfYm2xldm04el1pjcTE3K0JCYXc3Szh
sRUs1cXprWWVhcms5QThwN1AzR3pESytuZDNEUW93KzZVQzhTVk44Mm1ldjM4aw03TnRhWHRWMUNWcTZ
SZ3c0cGtzBwJkaTNidTJEZTdZZmFCQnhjcWZ2cVByVwPguU5UUTIybGZkVVZWVDY4c1RKS0Y1RG5TbVV
qZ2RxZzRtU1M5cG1zZkRKUjNHNlRvSDBpVzlhVjdMV0xIWVhLbGxURHQWTFRBdGtZSWFhbXAxUWpWdis
rdXlHVXhWZEowRE5WwFNtK2IxcVJ4cGw4NGRkZlgxTHAXTy9knj10c29kMHZzNWhHcmU5eHU4bytmcEx
SMWNHaE5URDZaNTdDOUtNV1hlZkpKt1o5NGJi0W9xZDFST25TN3FJVFR6SGltTXFpdmJPM2cwRGRWeWs
zV1FCaEJ6dEszNV1LTmRPbmM4TzNhY1M2ZkRaRmdLYVhMc0VKCDVyZHJsaUJxcDg5Y0pjcy9tN1R2czB
ya2pHZk40YjBrUG9abjNVSnVJT3JuWjIyeVAXzm12VXgrTzVnU3F1Y1YxbSt6U3VZTlZocTdUV2JEaUx
WdmxqcGxMbG9wNkNMWFArMnF0dkdMSUwvMXZpbUlTZE1CZ3pTb0ZaeXU2VHFkK2p6eGdzUGFWOUJDcWV
lL05qWws2djZsSz1jd2lVYy9TVHRmMUHEcE0zYjU5Mnk3aDNuAHg1b3pLNj1ITHBZV3VBd2FxUzVjdjI
2cTdjZWI4ZWZWWFszVAzaUZVOHPqMwtuU3daWEhNbW5Da1kwT2dhbG83VVFmU0NNM3FRUXIySC9YR1A
3c3NYeDQ1WWw5MUJ5ZUNlcDRtb1pvSCsxZkczeEQ0dFQ3eDhrd3lqOG53Yj1ldjI2VjBCNmQrN0g0ekt


```
E2V4YW1wbGVhbnBzZS5jb20xFDASBgNVBAoMC0V4YW1wbGUgT1JHMRAwDgYD
VQQLDAdFeGFtcGx1MQswCQYDVQGEwJVUzELMAkGA1UECAwCTVxkEjAQBgNVBAcM
CVdha2VmaWVsZDBZMBMGBYqGSM49AgEGCCqGSM49AwEHA0IABNQJs6wTqixc+S+V
DAajF1PNat10KEWJE5jcw0vm6qp09SDAAMZvb4HHRvs+P5YRpHrS1UPdvK+uEQbd
Wg31P9ujLDAqMAkGA1UdEwQCMAAwHQYDVr00BBYEFQsapcXV4ZovHAnRpPZwQe7
Yy20MAoGCCqGSM49BAMCA0gAMEUCIQc67za8EIuyRiKgNDXIP1s1aLr3jzH9WVXf
Hx4bJ+zCsGIgG/tVBut0JUUVvvoHIO/otAUACH5bNHP3uIziDS+PTUc=
-----END CERTIFICATE-----      -----BEGIN EC PRIVATE KEY-----
MHCcAQEEIFNpFhJvod3jKvbrLLzKTWKFxxaZ417kMchx3NyytQYUoAoGCCqGSM49
AwEHoUQDQGA1AmzrBOqLFz5L5UMBqMWU81q3XQoRYkTmNxY6+bqqk71IMAAxm9v
gceu+z4/1hGketKVQ928r64RBt1aDfU/2w==
-----END EC PRIVATE KEY-----
```

The root certificate to validate certificate path in the X5C is:

EXAMPLE: CERTIFICATE PATH ROOT CERTIFICATE

```
-----BEGIN CERTIFICATE-----
MIIGTCCBAGgAwIBAgIUdT9qLX0sVMRe8l0sLmHd3mZovQ0wDQYJKoZIhvcNAQEL
BQAwgZsxCzAdBgNVBAMMFkVYU1QTEUgTURTMjBURVNUIFJPT1QxIjAgBgkqhkiG
9w0BCQEWZ2V4YW1wbGVhbnBzZS5jb20xFDASBgNVBAoMC0V4YW1wbGUgT1JH
MRAwDgYDVQQLDAdFeGFtcGx1MQswCQYDVQGEwJVUzELMAkGA1UECAwCTVxkEjAQ
BgNVBAcMCVdha2VmaWVsZDAeFw0yMTA0MTkxMTM1MDdaFw00ODA5MDQxMTM1MDda
MIGbMR8wHQYDVQDDDBZFEFNUExFIE1EUzMGVEVTVCBST09UMSIwIAIJKoZIhvcN
AQkBFHnleGFtcGx1QGV4YW1wbGUuYy29tMRQwEgYDVQKDAFeGFtcGx1IE9SRzEQ
MA4GA1UECwwHRXhbnBzZTELMakGA1UEBhMCVVMxMzA1BgNVBAGMAk1ZMRlWFAyD
VQQHDA1XYWt1Zm1lbGQwggiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDD
jF5wyEWuhwDhsZosGdGFTCcI677rW881vV+UfW38J+K2ioFFNeGVsxbcebK6AV0i
CDPFj0974IpeD9SF0hWAHoDu/LCfXQWp8ZgQ91ULYWoW8o7NNSp01nbN9zma06/
xKNCa0bzjmXoGqglqnP1AtRcWYvXOSKZy1rcPeDv4Dhcdp6W72fBw0eWIqOhsrI
tuY2/N8ItBPiG03EX72nACq4nZJ/nAICuBER8STSFPPzvE97TvShsi1FD8a0611W
kR/QkreAGjMI++Gbb2Qc1nN9Y/VEDbMDhQtXQRPdFwubTjejkN9hK0tF3B71Yrw
Irnng3V9RoPMFdapWMzS1I+WWHog0oTj1PqWJDDg7+z1I6vSDeVWAMKr9mq1w10GN
zgBopIjd91RWkRtt2kQSPX9XxqS4E1gDDr8MKbpm3JuubQtNCg9D7Ljvzb6vwwUr
bPHH+oREvucsp0PZ5PpizloepGICLFxQQCuLGY2n7Ah10J0FXJq0FCaK3TWHwBv
ZsaY5DgBuUvdUrwrtgZNg2eg2omWXEepiVFQn3Fvj43Wh2npPMgIe5P0rwnCvR0x
acz4rtajKS1ucoB9b9iKqM2+M1y/FDIgVf1fWEHwK7YdzxMlgOeLdeV/kqRU5PE
U1LU9a2Ewd0ErrPbPKZmIfbs/L4B3k4zejMDH3Y+ZwIDAQAB01MwUTAdBgNVHQ4E
FgQU8sWwq1TrurK7xMTw01dKfeJBbCMwHwYDVR0jBBgwFoAU8sWwq1TrurK7xMTw
01dKfeJBbCMwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAgEAFw6M
1PiIfCPIBQ5EBUPNmRvRFuDPol0mDofnf/+mv63LqWQZAdo/W8tzZ9kOFhq24SiL
w0H7fsdG/jeREXiIZMNoW/ra6Uac8sU+FYF7Q+qp6CQL1SQbDcpVMifTQjcBk2xh
+aLK9SrrXBqnTAhwS+offGtAW8DpoLuH4tAcQmIjlgM1N65jnELCuqNR/wpA+zch
8LZW8saQ2cwRCwdr8mAzz0LbsDSVCHxQF3/kQjPT7Nao1q2iWcY30YcRmKrieHDP
67yeLubVmetfZis2d6Z1kqHLB4ZW1xX4otsEFkuTJA3HWDrsNyhTwx1YoCLsYut5
Zp0myqPNBq28w6qGMyyoJN0Z4RzMEO3R6i/MQNfhK55/802HciM6xb5t/aBSuHPK
1BDrfWphRnKYkaNtlUo35qV5IbKKGau3SdZdSRciaXUd/p81YmoF01U1hhMz/Rqr
```

```
1k2gyA0a9tF8+awCeanYt5izl8Y00Flr0U1SQ5UQw4szqqZqbrf4e8fRuU2TXNx4
zk+ImE7WRB44f6mSD746ZCBRogZ/SA5jUBu+OPe4/sEtERWRcQD+fXgce9ZEN0+p
eyJIKAsl5Rm2Bmggy5IoyWwSG5W+WekGyEokpslou2Yc6EjUj5ndZWz5EiHAIQ74
hNfDoCZIXVVLU3Qbp8a0S1bmsoT2J0sspIbtZUg=
-----END CERTIFICATE-----
```

3.2. Metadata BLOB object processing rules§

The FIDO Server MUST follow these processing rules:

1. Download and cache the root signing trust anchor from the respective MDS root location e.g. More information can be found at <https://fidoalliance.org/metadata/>
 2. To validate the digital certificates used in the digital signature, the certificate revocation information MUST be available in the form of CRLs at the respective MDS CRL location e.g. More information can be found at <https://fidoalliance.org/metadata/>
 3. The FIDO Server MUST be able to download the latest metadata BLOB object from the well-known URL when appropriate, e.g. <https://mds.fidoalliance.org/>. The `nextUpdate` field of the [Metadata BLOB](#) specifies a date when the download SHOULD occur at latest.
 4. If the `x5u` attribute is present in the JWT Header, then:
 1. The FIDO Server MUST verify that the URL specified by the `x5u` attribute has the same web-origin as the URL used to download the metadata BLOB from. The FIDO Server SHOULD ignore the file if the web-origin differs (in order to prevent loading objects from arbitrary sites).
 2. The FIDO Server MUST download the certificate (chain) from the URL specified by the `x5u` attribute [\[JWS\]](#). The certificate chain MUST be verified to properly chain to the metadata BLOB signing trust anchor according to [\[RFC5280\]](#). All certificates in the chain MUST be checked for revocation according to [\[RFC5280\]](#).
 3. The FIDO Server SHOULD ignore the file if the chain cannot be verified or if one of the chain certificates is revoked.
- The requirements for verifying certificate revocation, are only applicable to the MDS BLOB payload certificates. It is up to the server vendors whether to enforce CRL check for the certificates in the individual metadata statements.
5. If the `x5u` attribute is missing, the chain should be retrieved from the `x5c` attribute. If that attribute is missing as well, Metadata BLOB signing trust anchor is considered the BLOB signing certificate chain.
 6. Verify the signature of the Metadata BLOB object using the BLOB signing certificate chain (as determined by the steps above). The FIDO Server SHOULD ignore the file if the signature is

invalid. It SHOULD also ignore the file if its number (no) is less or equal to the number of the last Metadata BLOB object cached locally.

7. Write the verified object to a local cache as required.
8. Iterate through the individual entries (of type `MetadataBLOBPayloadEntry`). For each entry:
 1. Ignore the entry if the AAID, AAGUID or `attestationCertificateKeyIdentifiers` is not relevant to the relying party (e.g. not acceptable by any policy)
 2. Check whether the status report of the authenticator model has changed compared to the cached entry by looking at the fields `timeOfLastStatusChange` and `statusReport`.

Update the status of the cached entry. It is up to the relying party to specify behavior for authenticators with status reports that indicate a lack of certification, or known security issues. However, the status `REVOKED` indicates significant security issues related to such authenticators.

Authenticators with an unacceptable status should be marked accordingly. This information is required for building registration and authentication policies included in the registration request and the authentication request [\[UAFProtocol\]](#).

3. Update the cached metadata statement.

4. Considerations§

This section is not normative.

This section describes the key considerations for designing this metadata service.

Need for Authenticator Metadata

When defining policies for acceptable authenticators, it is often better to describe the required authenticator characteristics in a generic way than to list individual authenticator AAIDs. The metadata statements provide such information. Authenticator metadata also provides the trust anchor required to verify attestation objects.

The metadata service provides a standardized method to access such metadata statements.

Integrity and Authenticity

Metadata statements include information relevant for the security. Some business verticals might even have the need to document authenticator policies and trust anchors used for verifying attestation objects for auditing purposes.

It is important to have a strong method to verify and proof integrity and authenticity and the freshness of metadata statements. We are using a single digital signature to protect the integrity and authenticity of the Metadata BLOB object and all metadata statements.

Organizational Impact

The FIDO Alliance has control over the FIDO certification process and authentication vendors provide the metadata as part of that process. With this metadata service, the list of known authenticators and their metadata statements need to be updated, signed and published regularly. A single signature needs to be generated in order to protect the integrity and authenticity of the metadata BLOB object and all embedded metadata statements.

Performance Impact

Metadata BLOB objects and metadata statements can be cached by the FIDO Server.

The update policy can be specified by the relying party.

The metadata BLOB object includes a date for the next scheduled update. As a result there is *no additional impact* to the FIDO Server during FIDO Authentication or FIDO Registration operations.

High Security Environments

Some high security environments might only trust internal policy authorities. FIDO Servers in such environments could be restricted to use metadata BLOB objects from a proprietary trusted source only. The metadata service is the baseline for most relying parties.

Extended Authenticator Information

Some relying parties might want additional information about authenticators before accepting them. The policy configuration is under control of the relying party, so it is possible to only accept authenticators for which additional data is available and meets the requirements.

Index§

Terms defined by this specification§

[aaguid](#), in §3.1.1

[aaid](#), in §3.1.1

[attestationCertificateKeyIdentifiers](#), in §3.1.1

["ATTESTATION_KEY_COMPROMISE"](#), in §3.1.4

[AuthenticatorStatus](#), in §3.1.4

[authenticatorVersion](#), in §3.1.3

[BiometricStatusReport](#), in §3.1.2

[biometricStatusReports](#), in §3.1.1

[certificate](#), in §3.1.3

certificateNumber

[dict-member for BiometricStatusReport](#), in §3.1.2

[dict-member for StatusReport](#), in §3.1.3

certificationDescriptor

[dict-member for BiometricStatusReport](#), in §3.1.2

[dict-member for StatusReport](#), in §3.1.3

certificationPolicyVersion

[dict-member for BiometricStatusReport](#), in §3.1.2

[dict-member for StatusReport](#), in §3.1.3

certificationRequirementsVersion

[dict-member for BiometricStatusReport](#), in §3.1.2

[dict-member for StatusReport](#), in §3.1.3

[certLevel](#), in §3.1.2

[date](#), in §3.1.5

effectiveDate

[dict-member for BiometricStatusReport](#), in §3.1.2

[dict-member for StatusReport](#), in §3.1.3

[entries](#), in §3.1.6

["FIDO_CERTIFIED"](#), in §3.1.4

["FIDO_CERTIFIED_L1"](#), in §3.1.4

["FIDO_CERTIFIED_L1plus"](#), in §3.1.4

["FIDO_CERTIFIED_L2"](#), in §3.1.4

["FIDO_CERTIFIED_L2plus"](#), in §3.1.4

["FIDO_CERTIFIED_L3"](#), in §3.1.4

["FIDO_CERTIFIED_L3plus"](#), in §3.1.4

[legalHeader](#), in §3.1.6

[MetadataBLOBPayload](#), in §3.1.6

[MetadataBLOBPayloadEntry](#), in §3.1.1

[metadataStatement](#), in §3.1.1

[modality](#), in §3.1.2

[nextUpdate](#), in §3.1.6

[no](#), in §3.1.6

["NOT_FIDO_CERTIFIED"](#), in §3.1.4

["REVOKED"](#), in §3.1.4

[RogueListEntry](#), in §3.1.5

[rogueListHash](#), in §3.1.1

[rogueListURL](#), in §3.1.1

["SELF_ASSERTION_SUBMITTED"](#), in §3.1.4

[sk](#), in §3.1.5

[status](#), in §3.1.3

[StatusReport](#), in §3.1.3

[statusReports](#), in §3.1.1

[timeOfLastStatusChange](#), in §3.1.1

["UPDATE_AVAILABLE"](#), in §3.1.4

[url](#), in §3.1.3

["USER_KEY_PHYSICAL_COMPROMISE"](#), in §3.1.4

["USER_KEY_REMOTE_COMPROMISE"](#), in §3.1.4

["USER_VERIFICATION_BYPASS"](#), in §3.1.4

Terms defined by reference§

[webauthn-1] defines the following terms:

AAGUID

[WebIDL] defines the following terms:

DOMString

unsigned long

unsigned short

References§

Normative References§

[FIDOAuthenticatorSecurityRequirements]

Rolf Lindemann; Dr. Joshua E. Hill; Douglas Biggs. [FIDO Authenticator Security Requirements](https://fidoalliance.org/specs/fido-security-requirements/fido-authenticator-security-requirements-v1.4-fd-20201102.html). November 2020. Final Draft. URL: <https://fidoalliance.org/specs/fido-security-requirements/fido-authenticator-security-requirements-v1.4-fd-20201102.html>

[FIDOBiometricsRequirements]

Stephanie Schuckers; et al. [FIDO Biometrics Requirements](https://fidoalliance.org/specs/biometric/requirements/Biometrics-Requirements-v2.0-fd-20201006.html). October 2020. URL: <https://fidoalliance.org/specs/biometric/requirements/Biometrics-Requirements-v2.0-fd-20201006.html>

[FIDOMetadataStatement]

B. Jack; R. Lindemann; Y. Ackeremann. [FIDO Metadata Statements](https://fidoalliance.org/specs/mds/fido-metadata-statement-v3.0-ps-20210518.html). Proposed Standard. URL: <https://fidoalliance.org/specs/mds/fido-metadata-statement-v3.0-ps-20210518.html>

[JWS]

M. Jones; J. Bradley; N. Sakimura. [JSON Web Signature \(JWS\)](https://tools.ietf.org/html/rfc7515). May 2015. RFC. URL: <https://tools.ietf.org/html/rfc7515>

[JWT]

M. Jones; J. Bradley; N. Sakimura. [JSON Web Token \(JWT\)](https://tools.ietf.org/html/rfc7519). May 2015. RFC. URL: <https://tools.ietf.org/html/rfc7519>

[RFC4648]

S. Josefsson. [The Base16, Base32, and Base64 Data Encodings \(RFC 4648\)](http://www.ietf.org/rfc/rfc4648.txt). October 2006. URL: <http://www.ietf.org/rfc/rfc4648.txt>

[RFC5280]

D. Cooper; et al. [Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List \(CRL\) Profile](https://tools.ietf.org/html/rfc5280). May 2008. URL: <https://tools.ietf.org/html/rfc5280>

[WEBAUTHN-1]

Dirk Balfanz; et al. [Web Authentication: An API for accessing Public Key Credentials Level 1](https://www.w3.org/TR/webauthn-1/). 4 March 2019. REC. URL: <https://www.w3.org/TR/webauthn-1/>

[WebIDL]

Boris Zbarsky. [Web IDL](https://heycam.github.io/webidl/). 15 December 2016. ED. URL: <https://heycam.github.io/webidl/>

[WebIDL-ED]

Cameron McCormack. [Web IDL](http://heycam.github.io/webidl/). 13 November 2014. Editor's Draft. URL: <http://heycam.github.io/webidl/>

Informative References§

[FIDOEcdaaAlgorithm]

R. Lindemann; et al. [FIDO ECDA Algorithm](https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-ecdaa-algorithm-v2.0-id-20180227.html). Implementation Draft. URL: <https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-ecdaa-algorithm-v2.0-id-20180227.html>

[FIDOGlossary]

R. Lindemann; et al. [FIDO Technical Glossary](https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-glossary-v2.0-id-20180227.html). Implementation Draft. URL: <https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-glossary-v2.0-id-20180227.html>

[FIDOKeyAttestation]

FIDO 2.0: Key attestation format. URL: <https://fidoalliance.org/specs/fido-v2.0-ps-20150904/fido-key-attestation-v2.0-ps-20150904.html>

[ITU-X690-2008]

X.690: Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), (T-REC-X.690-200811). November 2008. URL: <https://www.itu.int/rec/T-REC-X.690-200811-S>

[RFC2119]

S. Bradner. [Key words for use in RFCs to Indicate Requirement Levels](https://tools.ietf.org/html/rfc2119). March 1997. Best Current Practice. URL: <https://tools.ietf.org/html/rfc2119>

[UAFProtocol]

R. Lindemann; et al. [FIDO UAF Protocol Specification v1.2](https://fidoalliance.org/specs/fido-uaf-v1.2-ps-20201020/fido-uaf-protocol-v1.2-ps-20201020.html). Proposed Standard. URL: <https://fidoalliance.org/specs/fido-uaf-v1.2-ps-20201020/fido-uaf-protocol-v1.2-ps-20201020.html>

IDL Index§

```
dictionary MetadataBLOBPayloadEntry {  
    AAID                aaaid;  
    AAGUID              aaguid;  
    DOMString[]         attestationCertificateKeyIdentifiers;  
    MetadataStatement   metadataStatement;  
    BiometricStatusReport[] biometricStatusReports;  
    required StatusReport[] statusReports;  
    required DOMString   timeOfLastStatusChange;  
    DOMString            rogueListURL;  
    DOMString            rogueListHash;  
};  
  
dictionary BiometricStatusReport {  
    required unsigned short certLevel;  
    required DOMString      modality;  
    DOMString               effectiveDate;  
    DOMString               certificationDescriptor;  
    DOMString               certificateNumber;  
    DOMString               certificationPolicyVersion;  
    DOMString               certificationRequirementsVersion;  
};
```

```

dictionary StatusReport {
    required AuthenticatorStatus status;
    DOMString effectiveDate;
    unsigned long authenticatorVersion;
    DOMString certificate;
    DOMString url;
    DOMString certificationDescriptor;
    DOMString certificateNumber;
    DOMString certificationPolicyVersion;
    DOMString certificationRequirementsVersion;
};

```

```

enum AuthenticatorStatus {
    "NOT_FIDO_CERTIFIED",
    "FIDO_CERTIFIED",
    "USER_VERIFICATION_BYPASS",
    "ATTESTATION_KEY_COMPROMISE",
    "USER_KEY_REMOTE_COMPROMISE",
    "USER_KEY_PHYSICAL_COMPROMISE",
    "UPDATE_AVAILABLE",
    "REVOKED",
    "SELF_ASSERTION_SUBMITTED",
    "FIDO_CERTIFIED_L1",
    "FIDO_CERTIFIED_L1plus",
    "FIDO_CERTIFIED_L2",
    "FIDO_CERTIFIED_L2plus",
    "FIDO_CERTIFIED_L3",
    "FIDO_CERTIFIED_L3plus"
};

```

```

dictionary RogueListEntry {
    required DOMString sk;
    required DOMString date;
};

```

```

dictionary MetadataBLOBPayload {
    DOMString legalHeader;
    required Number no;
    required DOMString nextUpdate;
    required MetadataBLOBPayloadEntry[] entries;
};

```