A message sent by the Issuer to the potential Holder, describing the credential they intend to offer and possibly the price they expect to be paid.

{

"@type": "https://didcomm.org/issue-credential/%VER/offer-credential",

"@id": "<uuid of offer message>",

"goal\_code": "<goal-code>",

"replacement\_id": "<issuer unique id>",

"comment": "<some comment>",

"multiple\_available": "<count>",

"credential\_preview": <json-ld object>,

"formats" : [

{

"attach\_id" : "<attach@id value>",

"format" : "<format-and-version>",

}

],

"offers~attach": [

{

"@id": "<attach@id value>",

"mime-type": "application/json",

"data": {

"base64": "<bytes for base64>"

}

}

],

"supplements": [

{

"type": "hashlink-data",

"ref": "<attachment identifier>",

"attrs": [{

"key": "field",

"value": "<fieldname>"

}]

},

{

"type": "issuer-credential",

"ref": "<attachment identifier>",

}

],

"~attach" : [] //attachments referred to in supplements

}

{

1. "@type": Indicates the type of the message, following the DIDComm conventions.
2. "@id": A universally unique identifier (UUID) assigned to the offer message.
3. "goal\_code": An optional field that indicates the goal of the message sender.
4. "replacement\_id": An optional field that helps coordinate credential replacement. If present and matching the replacement\_id of a previously issued credential, it informs the recipient that the offered credential is considered a replacement for the previous one.
5. "comment": An optional field that provides human-readable information about the credential offer, allowing evaluation by human judgment.
6. "multiple\_available": An optional field indicating the number of available verifiable credentials of the indicated type that the issuer can issue to the holder.
7. "credential\_preview": A JSON-LD object representing the data of the credential that the issuer is willing to issue. It conforms to the schema of a Credential Preview.
8. "formats": An array containing entries for each attachment in the "offers~attach" array. Each entry specifies the attachment's ID, format, and version.
9. "offers~attach": An array of attachments that further define the credential being offered. It may include additional information or clarify the formats or versions of the credential.
10. "supplements": An optional array of attachment descriptors that provide details about credential supplements. It can include information such as hashlink data or references to issuer credentials.
11. "~attach": Optional attachments related to the offered credential. Each attachment should be described in a "supplements" entry, referenced by its identifier.

This is a message sent by the potential Holder to the Issuer, to request the issuance of a credential.

{

"@type": "https://didcomm.org/issue-credential/%VER/request-credential",

"@id": "<uuid of request message>",

"goal\_code": "<goal-code>",

"comment": "<some comment>",

"formats" : [

{

"attach\_id" : "<attach@id value>",

"format" : "<format-and-version>",

}

],

"requests~attach": [

{

"@id": "<attachment identifier>",

"mime-type": "application/json",

"data": {

"base64": "<bytes for base64>"

}

},

],

"supplements": [

{

"type": "hashlink-data",

"ref": "<attachment identifier>",

"attrs": [{

"key": "field",

"value": "<fieldname>"

}]

},

{

"type": "issuer-credential",

"ref": "<attachment identifier>",

}

],

"~attach" : [] //attachments referred to in supplements

}

1. @type": Indicates the type of the message, following the DIDComm conventions.
2. "@id": A universally unique identifier (UUID) assigned to the request message.
3. "goal\_code": An optional field that indicates the goal of the message sender.
4. "comment": An optional field that provides human-readable information about the credential request, allowing evaluation by human judgment.
5. "formats": An array containing entries for each attachment in the "requests~attach" array. Each entry specifies the attachment's ID, format, and version.
6. "requests~attach": An array of attachments that define the requested formats for the credential. It specifies the attachments' identifiers, MIME types, and base64-encoded data.
7. "supplements": An optional array of attachment descriptors that provide details about credential supplements. It can include information such as hashlink data or references to issuer credentials.
8. "~attach": Optional attachments related to the requested credential. Each attachment should be described in a "supplements" entry, referenced by its identifier.

This message is sent by the issuer to issue a verifiable credential to the holder. It is sent in response to a valid "Request Credential" message received from the holder.

{

"@type": "https://didcomm.org/issue-credential/%VER/issue-credential",

"@id": "<uuid of issue message>",

"goal\_code": "<goal-code>",

"replacement\_id": "<issuer unique id>",

"comment": "<some comment>",

"more\_available": "<count>",

"formats" : [

{

"attach\_id" : "<attachment identifier>",

"format" : "<format-and-version>",

}

],

"credentials~attach": [

{

"@id": "<attachment identifier>",

"mime-type": "application/json",

"data": {

"base64": "<bytes for base64>"

}

}

],

"supplements": [

{

"type": "hashlink-data",

"ref": "<attachment identifier>",

"attrs": [{

"key": "field",

"value": "<fieldname>"

}]

},

{

"type": "issuer-credential",

"ref": "<attachment identifier>",

}

],

"~attach" : [] //attachments referred to in supplements

}

1. "@type": Indicates the type of the message, following the DIDComm conventions.
2. "@id": A universally unique identifier (UUID) assigned to the issue message.
3. "goal\_code": An optional field that indicates the goal of the message sender.
4. "replacement\_id": An optional field that provides an identifier used to manage credential replacement. If present and matching the replacement\_id of a previously issued credential, this new credential may be considered a replacement for the previous one. It is unique to the issuer and should not be used in a credential presentation.
5. "comment": An optional field that provides human-readable information about the issued credential, allowing evaluation by human judgment.
6. "more\_available": An optional field that, when set to a positive integer, indicates that the issuer has more instances of the verifiable credential type available for the holder. If not specified or set to 0, it means there are no more available credentials.
7. "formats": An array containing entries for each attachment in the "credentials~attach" array. Each entry specifies the attachment's ID, format, and version.
8. "credentials~attach": An array of attachments containing the issued credential in the requested format(s) specified by the holder. Each attachment includes an ID, MIME type, and base64-encoded data.
9. "supplements": An optional array of attachment descriptors that provide details about credential supplements. It can include information such as hashlink data or references to issuer credentials.
10. "~attach": Optional attachments related to the issued credential. Each attachment should be described in a "supplements" entry, referenced by its identifier.

Upon receiving the "Issue Credential" message, the holder's state moves to the appropriate state based on the presence of the "more\_available" field. If set to a positive integer, the issuer moves to the "offer-sent" state and waits for a "Request Credential" message from the holder. If not present or set to 0, the issuer moves to the "credential-issued" or "done" state, depending on whether the "~please-ack" decorator is included.

**CREDENTIAL PREVIEW**

{

"@type": "https://didcomm.org/issue-credential/%VER/credential-preview",

"attributes": [

{

"name": "<attribute name>",

"mime-type": "<type>",

"value": "<value>"

},

// more attributes

]

}

The main element of the object is the attributes array, which contains one or more attribute specifications. Each attribute specification is represented as an object with three properties: name, mime-type, and value.

1. name: This is a mandatory property that specifies the name of the attribute as a string. It identifies the type of data the attribute represents.
2. mime-type: This is an optional property that advises the issuer on how to render a binary attribute. It provides information on the type of the binary data using the MIME type format. If the mime-type property is missing, its value is null.
3. value: This is a mandatory property that holds the value of the attribute. The interpretation of the value property depends on the presence of the mime-type property:

* If mime-type is missing or null, the value is a string, similar to any other key-value pair in JSON.
* If mime-type is not null, the value is a base64url-encoded string representing a binary BLOB. The mime-type property provides information on how to interpret the BLOB after decoding it.

In summary, the preview credential object is used to construct a preview of the data for the credential that is to be issued. It allows specifying attributes with their names, values, and optional MIME types to provide guidance on rendering and interpreting binary attributes.

From a verifier to a prover, the request-presentation message describes values that need to be revealed and predicates that need to be fulfilled.

{

"@type": "https://didcomm.org/present-proof/%VER/request-presentation",

"@id": "<uuid-request>",

"goal\_code": "<goal-code>",

"comment": "some comment",

"will\_confirm": true,

"present\_multiple": false,

"formats" : [

{

"attach\_id" : "<attach@id value>",

"format" : "<format-and-version>",

}

],

"request\_presentations~attach": [

{

"@id": "<attachment identifier>",

"mime-type": "application/json",

"data": {

"base64": "<base64 data>"

}

}

]

}

1. @type: This field indicates the type of the message and specifies the version of the present-proof protocol being used.
2. @id: This field contains a universally unique identifier (UUID) for the request-presentation message.
3. goal\_code (optional): This field provides an optional code that indicates the goal or purpose of the message sender.
4. comment: This field allows the sender to provide additional human-readable information or comments about the request for a presentation.
5. will\_confirm (optional): This field indicates whether the verifier will send a post-presentation confirmation acknowledgment message. By default, it is set to false.
6. present\_multiple (optional): This field indicates whether the verifier wants the prover to send multiple presentations that satisfy the presentation request from different verifiable credentials. By default, it is set to false.
7. formats: This field is an array that contains one or more entries specifying the desired format and version for the verifiable presentation request. Each entry includes an attach\_id value, which corresponds to the identifier of an attachment, and a format value indicating the verifiable presentation request format and version.
8. request\_presentations~attach: This field is an array of attachments that contain the acceptable verifiable presentation requests. Each attachment includes an identifier (@id), mime type (mime-type), and data (base64) representing the attachment data in base64 format.

This is a response to a Presentation Request message and contains signed presentations.

{

"@type": "https://didcomm.org/present-proof/%VER/presentation",

"@id": "<uuid-presentation>",

"goal\_code": "<goal-code>",

"comment": "some comment",

"last\_presentation": true,

"formats" : [

{

"attach\_id" : "<attach@id value>",

"format" : "<format-and-version>",

}

],

"presentations~attach": [

{

"@id": "<attachment identifier>",

"mime-type": "application/json",

"data": {

"sha256": "f8dca1d901d18c802e6a8ce1956d4b0d17f03d9dc5e4e1f618b6a022153ef373",

"links": ["https://ibb.co/TtgKkZY"]

}

}

],

"supplements": [

{

"type": "hashlink-data",

"ref": "<attachment identifier>",

"attrs": [{

"key": "field",

"value": "<fieldname>"

}]

},

{

"type": "issuer-credential",

"ref": "<attachment identifier>",

}

],

"~attach" : [] //attachments referred to in supplements

}

1. @type: This field indicates the type of the message and specifies the version of the present-proof protocol being used.
2. @id: This field contains a universally unique identifier (UUID) for the presentation message.
3. goal\_code (optional): This field provides an optional code that indicates the goal or purpose of the message sender.
4. comment: This field allows the sender to provide additional human-readable information or comments about the presentation.
5. last\_presentation (optional): This field indicates whether this is the last presentation message to be sent in satisfying the presentation request. By default, it is set to true. If set to false, the prover must send another presentation message with additional presentation(s). The last presentation message from the prover must have a last\_presentation value of false.
6. formats: This field is an array that contains one or more entries specifying the format and version of the verifiable presentation. Each entry includes an attach\_id value, which corresponds to the identifier of an attachment, and a format value indicating the verifiable presentation format and version.
7. presentations~attach: This field is an array of attachments containing the presentation in the requested format(s). If the present\_multiple field is set to true in the request-presentation message from the verifier, the prover may include multiple proof presentations of the same format that satisfy the proof request. Here's a breakdown of the fields within the presentations~attach array:
8. @id: This field contains the identifier of the attachment. It serves as a unique reference for the attachment within the message.
9. mime-type: This field specifies the MIME type of the attachment. In this case, the mime-type is set to "application/json", indicating that the attachment data is in JSON format.
10. data: This field holds the actual data of the attachment. In the case of the presentation attachment, it contains the following sub-fields:

* sha256: This field contains the SHA-256 hash value of the attachment data. It provides a way to verify the integrity of the attachment.
* links: This field is an array of URLs that point to the location where the presentation data can be retrieved or accessed. The URLs serve as references or endpoints to retrieve the presentation information.

1. supplements: This field is an array of attachment descriptors that detail credential supplements. These supplements provide additional information related to the presented credentials. Each supplement entry includes a type, ref (attachment identifier), and optional attrs (key-value pairs).
2. ~attach: This field is an array of attachments related to the issued credential. Each attachment should be detailed in a supplements entry and referenced by the attachment id.

The use of attachments and supplements in the presentation message allows for the inclusion of supporting documents, evidence, or any other relevant information that helps verify the authenticity, integrity, or credibility of the presented credentials. These attachments can be referenced and used by the verifier to gain further insights, perform additional checks, or validate specific attributes or claims.

Data inside the presentation can include claims, proofs, contextual data, links, etc. Claims are the individual pieces of information (e.g., name, age, address) that are being presented by the prover. Each claim consists of a key-value pair. Proofs are cryptographic proofs or signatures that demonstrate the authenticity and integrity of the claims and the presentation as a whole. Contextual information can include additional metadata or contextual details related to the presentation, such as the timestamp of the presentation, the presentation format and version, the issuing authority, or any other relevant information that helps verify the presentation.

The presentation message is used to provide the verifier with signed presentations of the requested verifiable information. It may contain multiple presentations in different formats if specified in the request. The last\_presentation field indicates whether this is the final presentation or if additional presentations will follow. The message can be decorated with the ~please-ack decorator if the prover wants an acknowledgement that the presentation was accepted, or the verifier can indicate the will to send an acknowledgement using the will\_confirm property.