19.2. Informative References

[JWK.Thumbprint]	**Jones, M.**, "**JSON Web Key (JWK) Thumbprint**," draft-jones-jose-jwk-thumbprint (work in progress), July 2014 (**HTML**).
[OAuth.Post]	Jones, M. and B. Campbell, "**OAuth 2.0 Form Post Response Mode**," February 2014.
[OpenID.2.0]	OpenID Foundation, "OpenID Authentication 2.0," December 2007 (**TXT**, **HTML**).
[OpenID.Basic]	Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, "**OpenID Connect Basic Client Implementer's Guide 1.0**," November 2014.
[OpenID.Implicit]	Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, "**OpenID Connect Implicit Client Implementer's Guide 1.0**," November 2014.
[OpenID.PAPE]	**Recordon, D.**, **Jones, M.**, **Bufu, J., Ed.**, **Daugherty, J., Ed.**, and **N. Sakimura**, "OpenID Provider Authentication Policy Extension 1.0," December 2008 (**TXT**, **HTML**).
[OpenID.Session]	Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, "**OpenID Connect Session Management 1.0**," November 2014.
[RFC4949]	Shirey, R., "**Internet Security Glossary, Version 2**," RFC 4949, August 2007 (**TXT**).
[X.1252]	International Telecommunication Union, "**ITU-T Recommendation X.1252 -- Cyberspace security -- Identity management -- Baseline identity management terms and definitions**," ITU-T X.1252, November 2010.

Appendix A. Authorization Examples

TOC

The following are non-normative examples of Authorization Requests with different `response_type` values and their responses (with line wraps within values for display purposes only):

A.1. Example using response_type=code

TOC

```
GET /authorize?
  response_type=code
  &client_id=s6BhdRkqt3
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
  &scope=openid%20profile%20email
  &nonce=n-0S6_WzA2Mj
  &state=af0ifjsldkj HTTP/1.1
Host: server.example.com

HTTP/1.1 302 Found
Location: https://client.example.org/cb?
  code=Qcb0Orv1zh30vL1MPRsbm-diHiMwcLyZvn1arpZv-Jxf_11jnpEX3Tgfvk
  &state=af0ifjsldkj
```

A.2. Example using response_type=id_token

TOC

```
GET /authorize?
  response_type=id_token
  &client_id=s6BhdRkqt3
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
  &scope=openid%20profile%20email
  &nonce=n-0S6_WzA2Mj
  &state=af0ifjsldkj HTTP/1.1
Host: server.example.com

HTTP/1.1 302 Found
Location: https://client.example.org/cb#
  id_token=eyJraWQiOiIxZTlnZGs3IiwiYWxnIjoiUlMyNTYifQ.ewogImlz
```

```
  cyI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4
  Mjg5NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiAi
  bi0wUzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDEz
  MTEyODA5NzAsCiAibmFtZSI6ICJKYW5lIERvZSIsCiAiZ2l2ZW5fbmFtZSI6
  ICJKYW5lIiwKICJmYW1pbHlfbmFtZSI6ICJEb2UiLAogImdlbmRlciI6ICJm
  ZW1hbGUiLAogImJpcnRoZGF0ZSI6ICIwMDAwLTEwLTMxIiwKICJlbWFpbCI6
  ICJqYW5lZG9lQGV4YW1wbGUuY29tIiwKICJwaWN0dXJlIjogImh0dHA6Ly9l
  eGFtcGxlLmNvbS9qYW5lZG9lL21lLmpwZyIKfQ.rHQjEmBqn9Jre0OLykYNn
  spA10Qql2rvx4FsD00jwlB0Sym4NzpgvPKsDjn_wMkHxcp6CilPcoKrWHcip
  R2iAjzLvDNAReF97zoJqq880ZD1bwY82JDauCXELVR9O6_B0w3K-E7yM2mac
  AAgNCUwtik6SjoSUZRcf-O5lygIyLENx882p6MtmwaL1hd6qn5RZOQ0TLrOY
  u0532g9Exxcm-ChymrB4xLykpDj3lUivJt63eEGGN6DH5K6o33TcxkIjNrCD
  4XB1CKKumZvCedgHHF3IAK4dVEDSUoGlH9z4pP_eWYNXvqQOjGs-rDaQzUHl
  6cQQWNiDpWOl_lxXjQEvQ
  &state=af0ifjsldkj
```

The value of the `id_token` parameter is the ID Token, which is a signed JWT, containing three base64url encoded segments separated by period ('.') characters. The first segment represents the JOSE Header. Base64url decoding it will result in the following set of Header Parameters:

```
{"kid":"1e9gdk7","alg":"RS256"}
```

The `alg` value represents the algorithm that was used to sign the JWT, in this case `RS256`, representing RSASSA-PKCS-v1_5 using SHA-256. The `kid` value is a key identifier used in identifying the key to be used to verify the signature. If the `kid` value is unknown to the RP, it needs to retrieve the contents of the OP's JWK Set again to obtain the OP's current set of keys.

The second segment represents the Claims in the ID Token. Verifying and decoding the ID Token will yield the following Claims:

```
{
 "iss": "http://server.example.com",
 "sub": "248289761001",
 "aud": "s6BhdRkqt3",
 "nonce": "n-0S6_WzA2Mj",
 "exp": 1311281970,
 "iat": 1311280970,
 "name": "Jane Doe",
 "given_name": "Jane",
 "family_name": "Doe",
 "gender": "female",
 "birthdate": "0000-10-31",
 "email": "janedoe@example.com",
 "picture": "http://example.com/janedoe/me.jpg"
}
```

The third segment represents the ID Token signature, which is verified as described in **[JWS]**.

A.3. Example using response_type=id_token token

TOC

```
GET /authorize?
  response_type=id_token%20token
  &client_id=s6BhdRkqt3
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
  &scope=openid%20profile%20email
```

```
  &nonce=n-0S6_WzA2Mj
  &state=af0ifjsldkj HTTP/1.1
Host: server.example.com

HTTP/1.1 302 Found
Location: https://client.example.org/cb#
  access_token=jHkWEdUXMU1BwAsC4vtUsZwnNvTIxEl0z9K3vx5KF0Y
  &token_type=Bearer
  &id_token=eyJraWQiOiIxZTlnZGs3IiwiYWxnIjoiUlMyNTYifQ.ewogIml
  zcyI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ
  4Mjg5NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiA
  ibi0wUzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDE
  zMTEyODA5NzAsCiAiYXRfaGFzaCI6ICI3N1FtVVB0alBmeld0RjJBbnBLOVJ
  RIgp9.F9gRev0Dt2tKcrBkHy72cmRqnLdzw9FLCCSebV7mWs7o_sv2O5s6zM
  ky2kmhHTVx9HmdvNnx9GaZ8XMYRFeYk8L5NZ7aYlA5W56nsG1iWOou_-gji0
  ibWIuuf4Owaho3YSoi7EvsTuLFz6tq-dLyz0dKABMDsiCmJ5wqkPUDTE3QTX
  jzbUmOzUDli-gCh5QPuZAq0cNW3pf_2n4zpvTYtbmj12cVcxGIMZby7TMWES
  RjQ9_o3jvhVNcCGcE0KAQXejhA1ocJhNEvQNqMFGlBb6_0RxxKjDZ-Oa329e
  GDidOvvp0h5hoES4a8IuGKS7NOcpp-aFwp0qVMDLI-Xnm-Pg
  &state=af0ifjsldkj
```

Verifying and decoding the ID Token will yield the following Claims:

```
{
 "iss": "http://server.example.com",
 "sub": "248289761001",
 "aud": "s6BhdRkqt3",
 "nonce": "n-0S6_WzA2Mj",
 "exp": 1311281970,
 "iat": 1311280970,
 "at_hash": "77QmUPtjPfzWtF2AnpK9RQ"
}
```

A.4. Example using response_type=code id_token

TOC

```
GET /authorize?
  response_type=code%20id_token
  &client_id=s6BhdRkqt3
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
  &scope=openid%20profile%20email
  &nonce=n-0S6_WzA2Mj
  &state=af0ifjsldkj HTTP/1.1
Host: server.example.com

HTTP/1.1 302 Found
Location: https://client.example.org/cb#
  code=Qcb0Orv1zh30vL1MPRsbm-diHiMwcLyZvn1arpZv-Jxf_11jnpEX3Tgfvk
  &id_token=eyJraWQiOiIxZTlnZGs3IiwiYWxnIjoiUlMyNTYifQ.ewogIml
  zcyI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ
  4Mjg5NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiA
  ibi0wUzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDE
  zMTEyODA5NzAsCiAiY19oYXNoIjogIkxEa3RLZG9RYWszUGswY25YeENsdEE
  iCn0.XW6uhdrkBgcGx6zVIrCiROpWURs-4goO1sKA4m9jhJIImiGg5muPUcN
  egx6sSv43c5DSn37sxCRrDZZm4ZPBKKgtYASMcE20SDgvYJdJS0cyuFw7Ijp
  _7WnIjcrl6B5cmoM6ylCvsLMwkoQAxVublMwH10oAxjzD6NEFsu9nipkszWh
  sPePf_rM4eMpkmCbTzume-fzZIi5VjdWGGEmzTg32h3jiex-r5WTHbj-u5HL
  7u_KP3rmbdYNzlzd1xWRYTUs4E8nOTgzAUwvwXkIQhOh5TPcSMBYy6X3E7-_
  gr9Ue6n4ND7hTFhtjYs3cjNKIA08qm5cpVYFMFMG6PkhzLQ
  &state=af0ifjsldkj
```

Verifying and decoding the ID Token will yield the following Claims:

```
{
 "iss": "http://server.example.com",
 "sub": "248289761001",
 "aud": "s6BhdRkqt3",
 "nonce": "n-0S6_WzA2Mj",
 "exp": 1311281970,
 "iat": 1311280970,
 "c_hash": "LDktKdoQak3Pk0cnXxCltA"
}
```

A.5. Example using response_type=code token

TOC

```
GET /authorize?
  response_type=code%20token
  &client_id=s6BhdRkqt3
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
  &scope=openid%20profile%20email
  &nonce=n-0S6_WzA2Mj
  &state=af0ifjsldkj HTTP/1.1
Host: server.example.com

HTTP/1.1 302 Found
Location: https://client.example.org/cb#
  code=Qcb0Orv1zh30vL1MPRsbm-diHiMwcLyZvn1arpZv-Jxf_11jnpEX3Tgfvk
  &access_token=jHkWEdUXMU1BwAsC4vtUsZwnNvTIxEl0z9K3vx5KF0Y
  &token_type=Bearer
  &state=af0ifjsldkj
```

A.6. Example using response_type=code id_token token

TOC

```
GET /authorize?
  response_type=code%20id_token%20token
  &client_id=s6BhdRkqt3
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
  &scope=openid%20profile%20email
  &nonce=n-0S6_WzA2Mj
  &state=af0ifjsldkj HTTP/1.1
Host: server.example.com

HTTP/1.1 302 Found
Location: https://client.example.org/cb#
  code=Qcb0Orv1zh30vL1MPRsbm-diHiMwcLyZvn1arpZv-Jxf_11jnpEX3Tgfvk
  &access_token=jHkWEdUXMU1BwAsC4vtUsZwnNvTIxEl0z9K3vx5KF0Y
  &token_type=Bearer
  &id_token=eyJraWQiOiIxZTlnZGs3IiwiYWxnIjoiUlMyNTYifQ.ewogIml
  zcyI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ
  4Mjg5NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiA
  ibi0wUzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDE
  zMTEyODA5NzAsCiAiY19oYXNoIjogIkxEa3RLZG9RYWszUGswY25YeENsdEE
  iCn0.XW6uhdrkBgcGx6zVIrCiROpWURs-4goO1sKA4m9jhJIImiGg5muPUcN
  egx6sSv43c5DSn37sxCRrDZZm4ZPBKKgtYASMcE20SDgvYJdJS0cyuFw7Ijp
  _7WnIjcrl6B5cmoM6ylCvsLMwkoQAxVublMwH10oAxjzD6NEFsu9nipkszWh
```

```
 sPePf_rM4eMpkmCbTzume-fzZIi5VjdWGGEmzTg32h3jiex-r5WTHbj-u5HL
 7u_KP3rmbdYNzlzd1xWRYTUs4E8nOTgzAUwvwXkIQhOh5TPcSMBYy6X3E7-_
 gr9Ue6n4ND7hTFhtjYs3cjNKIA08qm5cpVYFMFMG6PkhzLQ
 &state=af0ifjsldkj
```

Verifying and decoding the ID Token will yield the following Claims:

```
{
 "iss": "http://server.example.com",
 "sub": "248289761001",
 "aud": "s6BhdRkqt3",
 "nonce": "n-0S6_WzA2Mj",
 "exp": 1311281970,
 "iat": 1311280970,
 "at_hash": "77QmUPtjPfzWtF2AnpK9RQ",
 "c_hash": "LDktKdoQak3Pk0cnXxCltA"
}
```

A.7. RSA Key Used in Examples

TOC

The following RSA public key, represented in JWK format, can be used to validate the ID Token signatures in the above examples (with line wraps within values for display purposes only):

```
{
 "kty":"RSA",
 "kid":"1e9gdk7",
 "n":"w7Zdfmece8iaB0kiTY8pCtiBtzbptJmP28nSWwtdjRu0f2GFpajvWE4VhfJA
      jEsOcwYzay7XGN0b-X84BfC8hmCTOj2b2eHT7NsZegFPKRUQzJ9wW8ipn_aD
      JWMGDuB1XyqT1E7DYqjUCEOD1b4FLpy_xPn6oV_TYOfQ9fZdbE5HGxJUzeku
      GcOKqOQ8M7wfYHhHHLxGpQVgL0apWuP2gDDOdTtpuld4D2LK1MZK99s9gaSj
      RHE8JDb1Z4IGhEcEyzkxswVdPndUWzfvWBBWXWxtSUvQGBRkuy1BHOa4sP6F
      KjWEeeF7gm7UMs2Nm2QUgNZw6xvEDGaLk4KASdIxRQ",
 "e":"AQAB"
}
```

Appendix B. Acknowledgements

TOC

As a successor version of OpenID, this specification heavily relies on ideas explored in **OpenID Authentication 2.0** [OpenID.2.0]. Please refer to Appendix C of OpenID Authentication 2.0 for the full list of the contributors for that specification.

In addition, the OpenID Community would like to thank the following people for their contributions to this specification:

Naveen Agarwal (naa@google.com), Google

Amanda Anganes (aanganes@mitre.org), MITRE

Casper Biering (cb@peercraft.com), Peercraft

John Bradley (ve7jtb@ve7jtb.com), Ping Identity

Tim Bray (tbray@textuality.com), Google

Johnny Bufu (jbufu@janrain.com), Janrain