Overview

• Draft: https://datatracker.ietf.org/doc/draft-spencer-oauth-claims/
• Lifts claims concept out of OpenID Connect (OIDC)
• Explains how to use in other, non-OIDC flows
• Stipulates claims I/O
• Adds extra examples not found in OIDC
• Defines clarifying terms
• Compatible with OIDC
Example

GET /authorize?
  client_id=s6BhdRkqt3&
  response_type=code&
  claims=%7B%0A%20%20%22access_token%22%20%3A%20%7B%0A%20%20%20%20%22https%3A%2F%2Fexample.com%2Fclaim1%22%20%3A
%20null%2C%0A%20%20%20%20%22fname%22%20%3A%20%7B%0A%20%20%20%20%22value%22%20%3A%20%22John%22%0A%20%20%20%20%20%20%20%20%22%0A%7D%0A%7D%0A%7D
Host: server.example.com
Example

GET /authorize?
   client_id=s6BhdRkqt3&
   response_type=code&
   claims={
      "access_token" : {
         "https://example.com/claim1" : null,
         "fname" : {
            "value" : "John" }
      }
   }

Host: server.example.com
Terms

• Claim, claim name, claim value from JWT
• Essential claims similar to OIDC except not tied to end user
• Others defined in draft
  • Critical claims are those required by client
  • Claims sink, claims request object, claims sink query object, etc.
Claims Sink

- Where client would like AS to put claims
- Examples
  - access_token
  - ?
  - *
  - id_token (in OIDC not draft)
  - userinfo (in OIDC not draft)
Requesting Claims

• A query for certain claims
• Ask that claims be put in certain claims sink
• Can request a certain value or values
• Values specified in preferred order

```json
{
  "access_token" : {
    "accountId" : {
      "values" : [
        "act-123", "act-456"],
      "essential" : true },
    "paymentId" : {
      "value" : "pid-123456",
      "essential" : true
    }
  }  
}
```
Essential Claims

• Essential to smooth authorization of tasks requested by RO
  • Not required
  • Not dictating AS assert something
• AS must not generate an error if not available

```json
{
  "access_token": {
    "accountId": {
      "values": ["act-123", "act-456"],
      "essential": true
    },
    "paymentId": {
      "value": "pid-123456",
      "essential": true
    }
  }
}
```
Critical Claims

• AS must return error if it doesn’t understand claims requested
• crit member of claims request obj:
  • Is a list of JSON pointers
  • Is like crit in JWT header
• JSON pointer defines how to escape slash in claim name

```json
{
  "crit" : [
    "access_token/verified_claims/verification/trust_framework/value",
    "access_token" : {
      "verified_claims" : {
        "verification" : {
          "trust_framework" : {
            "value" : "de_aml"
          }
        }
      }
    }
  ]
}
```
Special Claims Sinks

• ? – Client doesn’t care where claims end up
• * – Client wants all claims in all supported claim sinks
• Resource indicator – Client wants claims for certain RS (TBD)
Flows

• Claims request/response profiled for following authorization flows:
  • Code
  • Implicit
  • ROPC
  • CC
• Token refresh
• Token introspection
• Token exchange (TBD)
Authorization Flows

- **Code & Implicit**
  - Request is like OIDC using claims request parameter
  - Response includes space-separated list of granted claim names

- **ROPC and CC**
  - Claims request parameter (as code & implicit) & like scope request parameter
  - Response includes space-separated list of granted claim names

- **Error responses**
  - Maintains compatibility with OIDC
  - Defines more informative optional errors
Refresh

• Can send claims request parameter to down-scope AT
• Can be up-scope again if it doesn’t exceed original grant
• Difficulties to implement with regard to:
  1. Using scopes & claims together
  2. Policy changes at AS between time of grant and down-scope
    Both cases are out of scope and left to implementations or profiles
Token Introspection

• Response includes space-separated list of claim names that were authorized
Authorization Server Metadata

- claims_parameter_supported (same as OIDC)
- claims_supported (same as OIDC)
- critical_claims_supported
  - true / false (default) if critical claims are supported
  - Helpful when determining if AS/OP supports this draft
Open Questions

• Drop essential and leave that to OIDC?
• Restructure to avoid redundancy?
• How to integration with resource indicators?
• Inputs and suggestions on use with token exchange?
Next Version of Draft

• Finish writing:
  • Token exchange subsections
  • Resource indicator tie in
  • Privacy considerations
  • Security considerations
  • IANA considerations

• Add section about registration metadata, so client can register certain claims
Full Disclosure

• Curity support all of this in our product
  • Most is required by OIDC
    • Other aspects help make claim useful in practice
• We have no patents on any of these things
Request of WG

• Ask that WG adopt this draft as a work item