cornerstone

White paper

Cornerstone API

Architectural overview

by
CTO Dmitry Gerasimov, Ph.D.
and
CEO Anders Torvill Bjorvand, M.Sc./MBA

Version 1.0, 2019-02-08, © Kommunion AS 2019, support@kommunion.com

Cornerstone API is based on GraphQL which is significantly more flexible and elegant than the traditional REST API.

GraphQL

GraphQL is an open-source data query and manipulation language for APIs, and a runtime for fulfilling queries with existing data. GraphQL was developed internally by Facebook and later moved out and made public.

Key benefits:

- **No Overfetching**. The client specifies exactly which fields to fetch from the server. For example: fetch just list of names from the people database in one call and just phone numbers in another call to the same node.
- **Efficient requests**. The client can fetch different data with different parameters in a single API call.
- **Introspective**. A client can query the schema for details about the schema. Schema has all the information about data available through the GraphQL API, it is perfect for auto-generating API documentation.
- **Hierarchical**. API can be organized into a simple and understandable graph schema. The shape of a GraphQL call mirrors the shape of the JSON data it returns.
- **Strongly Typed**. Being strongly-typed makes GraphQL less error prone, can be validated during compile-time and can be used for supportive IDE/editor integrations such as auto-completion and validation.
- **No API versioning.** It is possible to deprecate the API on a field level. This makes it possible to evolve a GraphQL API over time without the need for versioning.
- **Growing ecosystem**. Community, libraries, etc.

Read more about GraphQL at graphql.org

Endpoint

Cornerstone API has one single endpoint:

```
https://your_domain.com/_graphql
```

Cornerstone basic structure

EXPAND ON:
Internally Cornerstone is Rooms -> Modules -> Objects
Cornerstone API follows this logic and forms the matching GraphQL hierarchy.

Schema overview

The two types of allowed operations in Cornerstone API are queries and mutations. Query is a way to retrieve data. Mutation is a way to manipulate data - create, edit, delete, take actions.

The best way to walk through the schema is to use Documentation explorer in GraphiQL Chrome extension. Point it to https://cornerstoneplatform.com/_graphql

Communicating with GraphQL

Query example

```
{
  room {
    module {
      people(id: 856) {
        people(filters: [{type: "text", value: "Anders"}]) {
          totalCount
          items {
            id
            nameFirst
            nameLast
          }
        }

      }
    }
  }
}
```

Response

```
{
  "data": {
    "room": {
      "module": {
        "people": {
          "people": {
            "totalCount": 1,
            "items": [
              {
                "id": 478,
                "nameFirst": "Anders",
                "nameLast": "Bjorvand"
              }            ]
          }
        }
      }
    }
  }
}
```

Authentication

In order to access Cornerstone API the client needs to pass a valid session key through a cookie named “CSNSESSID”. A session ID can be obtained by passing username and password to the “authentication” mutation.