Swagger Workshop Companion
Api Strategy Conference

Tony Tam

@fehguy

Source Code

https://github.com/fehguy/apistrat

- Use this document in case you fall behind!
- Listen to the talking points!

API Design with Swagger

- `cd design`
- `npm install`
- `node static-server.js`

Then…

- `http://localhost:8000/swagger-ui`

API Design with Swagger

- Key Points
  - Design with JSON before code
  - Visualize entire API
  - Share, collaborate & iterate
  - Make *this* your contract

Server Development

- Script file
  - pom.xml
- Mustache templates:
  - src/main/templates
- Swagger spec:
  - http://localhost:8000/docs/api-docs.json

Java Server

- `cd server`
- `mvn scala:run -Dlauncher=java-server`
- `cd java-server`
- `mvn package jetty:run`

Then in swagger-ui…

- `http://localhost:8002/api/api-docs`

Scala Server

- `cd server`
- `mvn scala:run -Dlauncher=scala-server`
- `cd scala-server`
- `./sbt`

  `=> container:start`

Then in swagger-ui…

- `http://localhost:8080/api-docs`

Node.js server

- `cd server`
- `mvn scala:run -Dlauncher=node-server`
- `cd nodejs`
- `node App/main.js`

Then in swagger-ui…

- `http://localhost:8002/api-docs.json`

Server Implementation with Swagger

- Key Points
  - Templates provided are a *starting* point
  - Easy to add languages
  - Follow your own lang/style/library
  - The contract defines the plumbing

Client Development

- Script file
  - pom.xml
- Mustache templates:
  - src/main/templates
- Swagger spec:
  - http://localhost:8000/docs/api-docs.json

Java Client

- `cd client`
- `mvn scala:run -Dlauncher=java-client`
- `cd java-client`
- `mvn package`

Then in your app…

- `EventsApi api = new EventsApi();`
- `Event event = api.findEventById(123);`

iOS Client

- `cd client`
- `mvn scala:run -Dlauncher=objc-client`
- `cd objc-client`
- `# create Sample project`
- `pod install`

Then in your app…

- `=> next page`

iOS Client

```
SWGEventsApi * api =[[SWGEventsApi alloc] init];
[api findEventByIdWithCompletionBlock:@1
                    completionHandler:
    ^(SWGEvent *event, NSError *error) {
        if(error) {
           // handle it
        }
        else {
           // do something with the SWGEvent
           NSLog(@"%@", [event asDictionary]);
        }
    }
];
```

Javascript Client

```
npm install swagger-client
node
> var client = require("swagger-client")
> var swagger = new client.SwaggerApi("http://localhost:8080/api-docs")
> swagger.build()
> swagger.apis.events.findEventById({eventId: 123});
```

Javascript Client (cont)

```
var client = require("swagger-client")

var swagger = new client.SwaggerApi(
  {
    url: "http://localhost:8080/api-docs",
    success: function() {
      if(swagger.ready === true) {
         swagger.apis.events.findEventById(
           {eventId: 1}
         );
      }
    }
  }
)
```

Javascript Client (Browser)

```
<script src='lib/swagger.js' type='text/javascript'></script>
<script type="text/javascript">
$(function () {
  var swagger = new window.SwaggerApi(
  {
    url: "http://localhost:8080/api-docs",
    success: function() {
      if(swagger.ready === true) {
        swagger.apis.events.findEventById(
        {
          eventId: 1
        },
// ...
```

Client Development using Swagger

- Key Points
  - Swagger JSON is machine readable
  - Templates provided are a *starting* point
  - Contract lets client & server develop in parallel
  - Not everyone needs to be a plumber

Further Info

- Swagger Wiki
  - https://github.com/wordnik/swagger-core/wiki
- Swagger IRC
  - irc.freenode.net #swagger
- Swagger mailing list
  - https://groups.google.com/forum/#!forum/swagger-swaggersocket
- Swagger in action
  - http://helloreverb.com