PV239 - 06 API

QUESTIONS

- What types of storage did we talk about in the previous exercise?

- What can we use Preferences for?

- What can we use Secure Storage for?

- What can we use SQLite for?

- How does SQLite database work on a phone?

GOALS

- Web API concepts

- Client-server communication

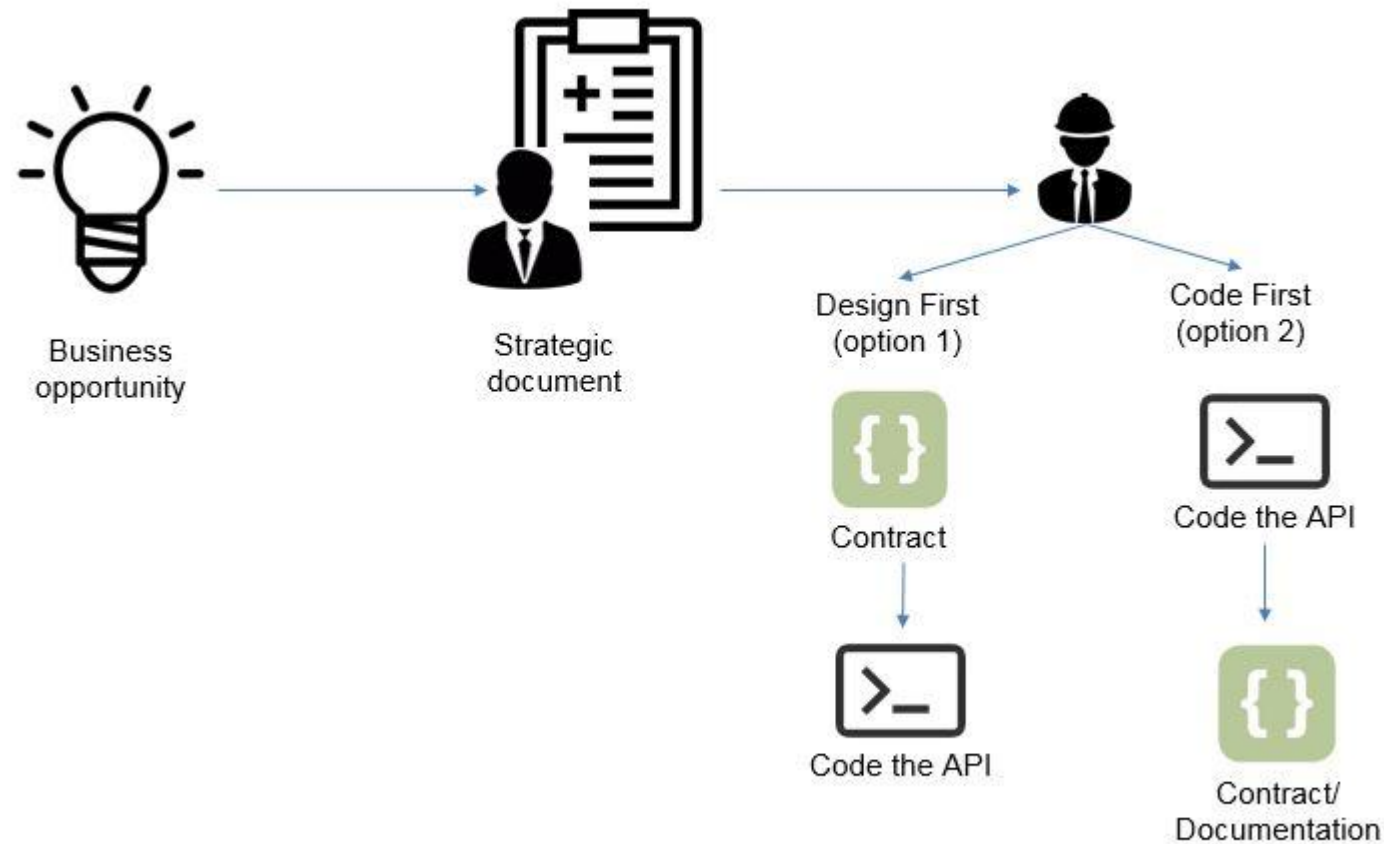- Generating API client and it's usage to communicate with API

OPENAPI

- Open specification

- Standard for describing REST API

- Language and framework independent

- YAML/JSON

- OpenAPI map: https://openapi-map.apihandyman.io/

SWAGGER

- Tools for creating and working with OpenAPI specification

- Swagger Editor
  - YAML/JSON

- Swagger CodeGen
  - Generating client/server part

- Swagger UI
  - Interactive documentation

APPROACHES FOR CREATING API SPECIFICATION

DESIGN FIRST APPROACH

- SwaggerHub
  - https://app.swaggerhub.com/

- Portal for design first approach

- Supports hosting on own server

CODE FIRST APPROACH

- ASP .NET Core

- Library **NSwag.AspNetCore**

OPEN API
CODE FIRST

Demo

SETUP

- Nuget package **NSwag.AspNetCore**

- Add configuration:

```
services.AddOpenApiDocument(document =>
{
    document.Title = "CookBook API";
    document.DocumentName = "cookbook-api";
});
```

SETUP

- Add middleware:

```
app.UseOpenApi();
```

- Add middleware for Swagger UI:

```
app.UseSwaggerUi3(settings =>
{
  settings.SwaggerRoutes.Add(
    new SwaggerUi3Route("CookBook API", "/swagger/cookbook-api/swagger.json"));
});
```

SWAGGER EDITOR

- Online version https://editor.swagger.io

- Support for running locally
  - Docker image

GENERATING – ONLINE

• Generator from Swaggeru

• Enables generating directly from browser

https://generator.swagger.io/

• Settings:

```
{
    "options": { "packageName": „LibraryName" },
    "swaggerUrl": „URL of OpenAPI specification"
}
```

GENERATING – SWAGGER CODEGEN CLI

- CLI tool

- Java app

- Download .jar file from Maven

GENERATING - AUTOREST

- CLI tool https://github.com/Azure/autorest

- Installed using NPM

- AutoRest.Csharp – generator for C#

GENERATING - NSWAGSTUDIO

- NSwag - CLI tool

- NSwagStudio - Desktop app

- https://github.com/RSuter/NSwag

- Settings:
  - Namespace
  - Set Operation Generation Mode
  - Path to output file

NSWAGSTUDIO

Demo

USING GENERATED CLIENT

Demo

ADDING CLIENT TO MOBILE APP

- Add client registration to DI container:

```
serviceCollection.AddHttpClient<IIngredientsClient,
IngredientsClient>(client =>

{

    client.BaseAddress = apiBaseUri;

});
```

- For usage in emulators
  - URL for „Localhost": 10.0.2.2

PROCESSORS

- Used for modifying the OpenAPI specification

- Can be used on multiple levels:
  - IOperationProcessor – modifies individual operations
  - ISchemaProcessor – modifies the schema
  - IDocumentProcessor – modifies the entire document