

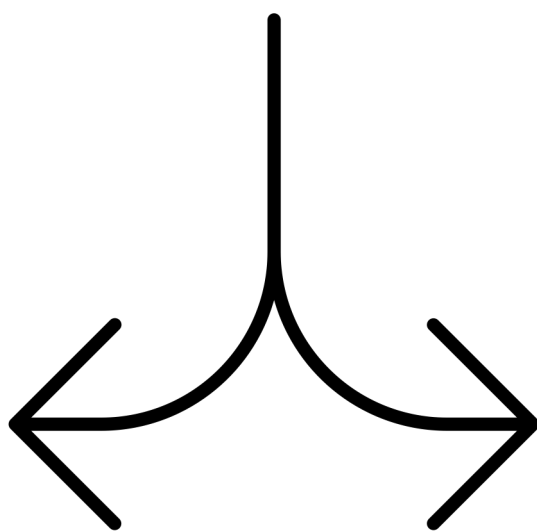
Problem

Syntactic description of offered REST API is required



Documentation and communication purpose

Outdated, inconsistent documentation, miscommunication



Generation of client-side API in language of choice

Imposed duplication, Incomplete API

Manual process: error-prone, time consuming

OpenAPI specification (OAS) defines a standard, language-agnostic interface to RESTful APIs.

Solution



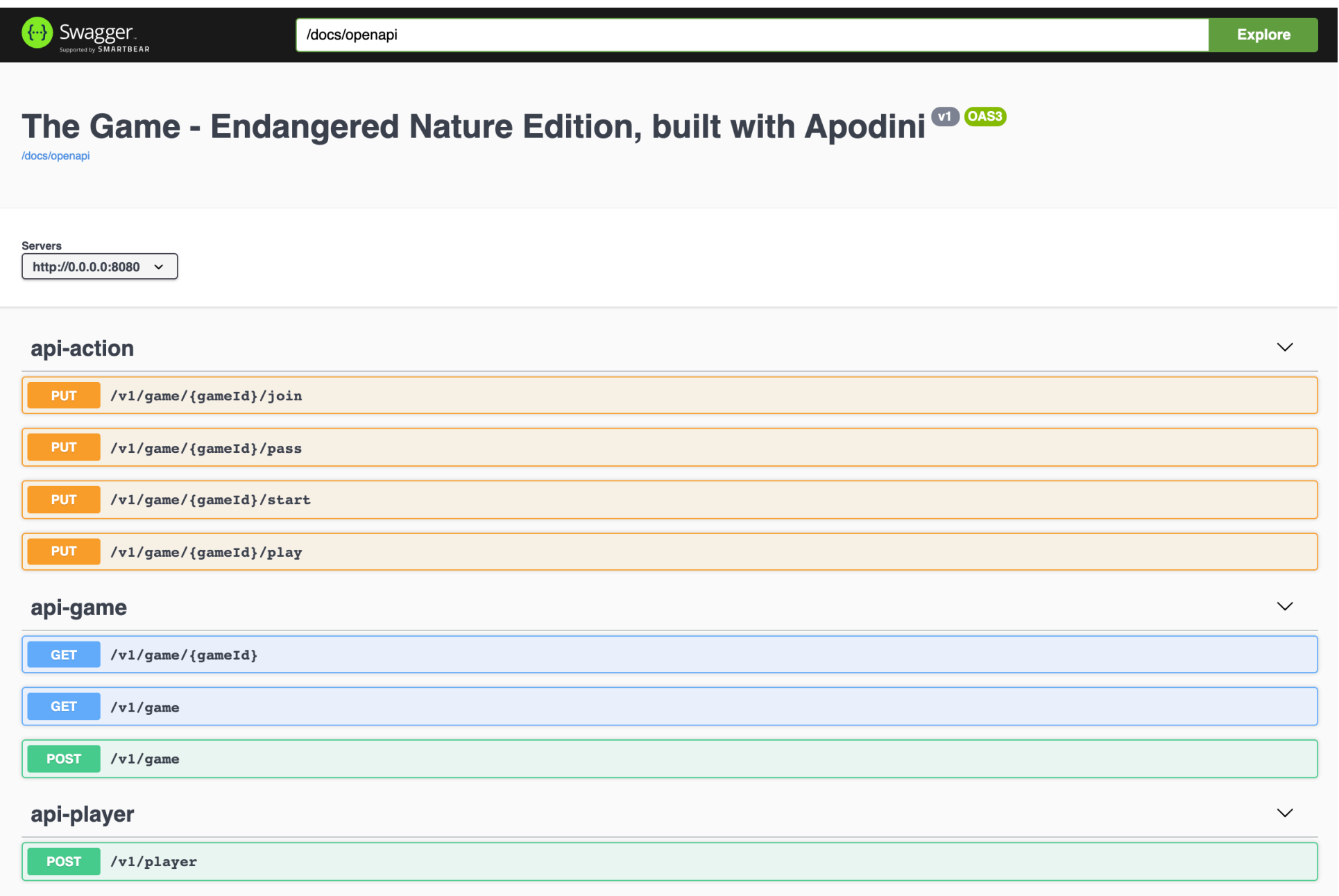
Automatic creation of **OAS** from **Apodini** enables developers to always having an accurate **documentation** of exposed **REST API** make use of **client code generation** tools.

[doc/openapi](#)

OAS 3.0

```
{
  "openapi": "3.0.0",
  "info": {
    "title": "The Game - Endangered Nature Edition, built with Apodini",
    "version": "v1"
  },
  "servers": [ { "url": "http://0.0.0.0:8080" } ],
  "paths": {
    "/v1/game": {
      "get": { "description": "Returns all games.", ... },
      "post": { "description": "Creates a new game with a specified name as query parameter.", ... },
      "/v1/game/{gameId}": {
        "get": {
          "description": "Returns the game having the id given in path.",
          "operationId": "GetGame",
          "responses": {
            "200": {
              "content": {
                "application/json": {
                  "schema": {
                    "$ref": "#/components/schemas/GameResponse"
                  }
                }
              },
              "description": "OK"
            }
          }
        }
      }
    }
  }
}
```

[ui/swagger](#)



Output on dedicated endpoints of Apodini web app

[apiGameService.ts](#)

```
@Injectable()
export class ApiGameService {

  /**
   * Creates a new game with a specified name as query parameter.
   * @param name @Parameter var name: String
   * @param observe set whether or not to return the data Observable as the body, response or events. defaults to returning the body.
   * @param reportProgress flag to report request and response progress.
   */
  public createGame(name?: string, observe?: 'body', reportProgress?: boolean): Observable<GameResponse>;
  public createGame(name?: string, observe?: 'response', reportProgress?: boolean): Observable<HttpResponse<GameResponse>>;
  public createGame(name?: string, observe?: 'events', reportProgress?: boolean): Observable<HttpEvent<GameResponse>>;
  public createGame(name?: string, observe: any = 'body', reportProgress: boolean = false ): Observable<any> {

    let queryParams = new HttpParams({encoder: new CustomHttpUrlEncodingCodec()});
    if (name !== undefined && name !== null) {
      queryParams = queryParams.set('name', <any>name);
    }

    let headers = this.defaultHeaders;

    // to determine the Accept header
    let httpHeaderAccepts: string[] = [
      'application/json'
    ];
    const httpHeaderAcceptSelected: string | undefined = this.configuration.selectHeaderAccept(httpHeaderAccepts);
    if (httpHeaderAcceptSelected !== undefined) {
      headers = headers.set('Accept', httpHeaderAcceptSelected);
    }

    // to determine the Content-Type header
    const consumes: string[] = [
    ];

    return this.httpClient.request<GameResponse>('post', `${this.basePath}/v1/game`,
      {
        params: queryParams,
        withCredentials: this.configuration.withCredentials,
        headers: headers,
        observe: observe,
        reportProgress: reportProgress
      }
    );
  }
}
```

Generated client SDK