

Metodika pre štruktúrovanie API

Patrik Papšo

November 2017

1 Štruktúrovanie API

Na komunikáciu so serverom systému iBazáru používame štandard *REST-API*, ktorý zahŕňa operácie *DELETE*, *GET*, *HEAD*, *POST* a *PUT*. Dáta v dopyte aj odpovedi musia byť vo formáte JSON. Po obdržaní dopytu server odpovie štandardizovanými HTTP odpoveďami spolu s dátami:

- **1xx** Informačná odpoveď.
- **2xx** Úspech.
- **3xx** Presmerovanie.
- **4xx** Chyba na strane klienta.
- **5xx** Chyba na strane servera.

1.1 URL

URL, ktoré počúvajú na dopyty je nutné deliť podľa logických celkov, ktoré v Django frameworku predstavujú aplikácie, t.j. aplikácia, ktorá sa stará o používateľov (*/users*) bude používať prefix */users* aj v URL dopytu.

Príklady štruktúry URL:

- GET */users/*
- POST */users/*
- GET */items/:slug:¹/*

¹Premenná *:slug*: predstavuje unikátny identifikátor inzerátu, ktorý bol vygenerovaný z názvu daného inzerátu

1.2 Štruktúra odpovedí API

Štruktúra odpovedí silno závisí od danej aplikácie, ktorá sa o túto odpoveď stará. Odpoveď by mala obsahovať prislúchajúcu HTTP hlavičku so správnym HTTP Status-om, dáta v odpovedi musia byť v korektnom JSON formátovaní. Ak odpoveď v dátach obsahuje odkaz na inú tabuľku, musí byť tento kľúč nahradený celým objektom z danej tabuľky. Príklad takéhoto nahradenia vo formáte JSON:

```
1 GET /items/test {
2   "id": ...,
3   "owner": {
4     "id": ...,
5     "username": ...,
6     "groups": [],
7     "userprofile": {
8       "city": {
9         "id": ...,
10        "name": ...,
11        "country": ...
12      }
13    }
14  },
15  "title": ...,
16  "city": {
17    "id": ...,
18    "name": ...,
19    "country": ...
20  },
21  "photos": [
22    {
23      "id": ...,
24      "created_at": ...,
25      "position": ...,
26      "description": ...,
27      "image": ...
28    }
29  ],
30  "created_at": ...,
31  "price": ...,
32  "slug": "test",
33  "description": ...
34 }
```