

SWAGGER UI

SWAGGER UI

Make your API Swagger

Umm, swagger?....

urban

DICTIONARY

look up any word, like surfboard:

swagger

search

word of the dayword of the yearfavoritesdictionaryeditthesaurusnamesmediastoreaddblog

randomA B C D E F G H I J K L M N O P Q R **S** T U V W X Y Z # new

trending

sapiosexual
bae
poopsterbate
tex-sex
tittybong
adults chewing gum
surfboard
fuck
catfish
bye felicia

categories

gaming
sports
food
sex
tv
film
celebrities
military
music
weather
insults

alphabetical

Swaggcabulary
Swagg Clown
Swagg Daddy
swagg'd out
Swagg'd up
Swagged
swagged on ya face
Swagged out
Swagged out till your
bags out
Swagged out to the
Maximum

1. swagger

How one presents him or her self to the world. Swagger is shown from how the person handles a situation. It can also be shown in the person's walk.

Ryan: "Denzel Washington has swagger in all his movie rolls"
Phil: "You can't forget about Al Pacino! He had swagger all over in Scarface!!!"

by [Ryan Jackson](#) February 19, 2005

7005 3047

  Buy "swagger" mugs & shirts >

Words related to **swagger**:

swag cool swagg swagga style
awesome fresh sexy yolo fly pimp

unexpectedcasting.com

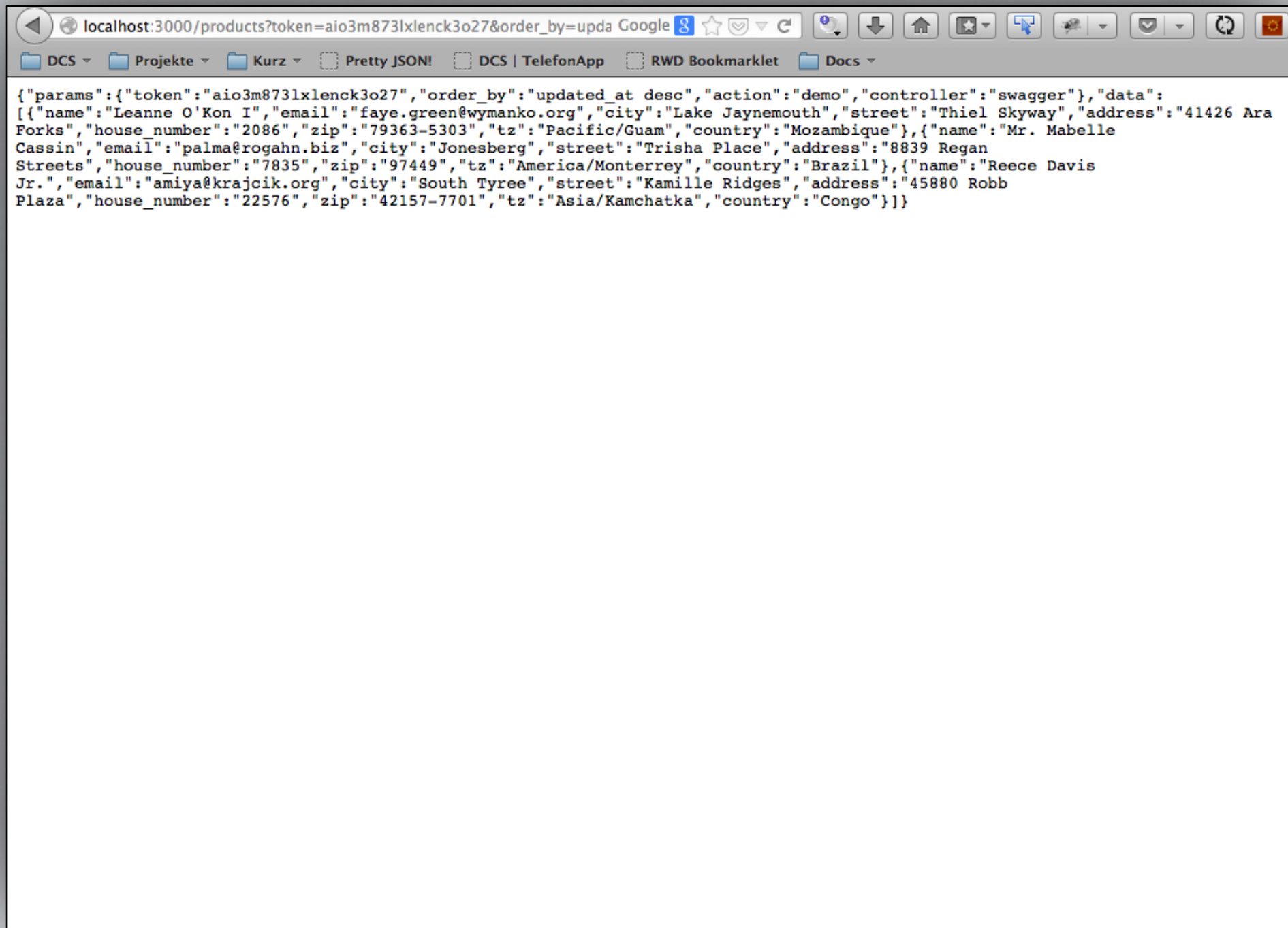
ÜBERRASCHE UNS
JETZT!

ALCATEL
onetouch.

Mmmm, Swagger:

- Pretty UI for your REST API
- Documentation
- Quick Test & Debugging Tool

Going from this...



The screenshot shows a web browser window with the address bar displaying `localhost:3000/products?token=aio3m873lxlenc3o27&order_by=upda`. The browser's bookmark bar includes links to "DCS", "Projekte", "Kurz", "Pretty JSON!", "DCS | TelefonApp", "RWD Bookmarklet", and "Docs". The main content area displays a JSON response from a REST client, which includes a "params" object and a "data" array of three user objects. Each object contains fields for name, email, city, street, address, house_number, zip, tz, and country.

```
{
  "params": {
    "token": "aio3m873lxlenc3o27",
    "order_by": "updated_at desc",
    "action": "demo",
    "controller": "swagger"
  },
  "data": [
    {
      "name": "Leanne O'Kon I",
      "email": "faye.green@wymanko.org",
      "city": "Lake Jaynemouth",
      "street": "Thiel Skyway",
      "address": "41426 Ara Forks",
      "house_number": "2086",
      "zip": "79363-5303",
      "tz": "Pacific/Guam",
      "country": "Mozambique"
    },
    {
      "name": "Mr. Mabelle Cassin",
      "email": "palma@rogahn.biz",
      "city": "Jonesberg",
      "street": "Trisha Place",
      "address": "8839 Regan Streets",
      "house_number": "7835",
      "zip": "97449",
      "tz": "America/Monterrey",
      "country": "Brazil"
    },
    {
      "name": "Reece Davis Jr.",
      "email": "amiya@krajcik.org",
      "city": "South Tyree",
      "street": "Kamille Ridges",
      "address": "45880 Robb Plaza",
      "house_number": "22576",
      "zip": "42157-7701",
      "tz": "Asia/Kamchatka",
      "country": "Congo"
    }
  ]
}
```

To this...

The screenshot shows a web browser window with the URL `localhost:3000/docs/#!/products/find_all`. The browser's address bar and tabs are visible at the top. The main content area features the **docSwagger** logo and three API sections: **sessions : User session operations**, **products : Data set operations**, and **dev : Development & testing operations**. Each section lists API endpoints with their HTTP methods, paths, and descriptions. The **sessions** section includes `POST /sessions` and `DELETE /sessions/{token}`. The **products** section includes `POST /products`, `GET /products`, `GET /products/{id}`, `GET /products/{id}/details`, `PATCH /products/{id}`, and `DELETE /products/{id}`. The **dev** section includes `PATCH /dev/reset_db`. At the bottom, a footer indicates the base URL and API version: `[BASE URL: http://localhost:3000/docs/ , API VERSION: 1.0]`.

docSwagger

sessions : User session operations [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

- POST** `/sessions` Create a new session and return auth token (destroy old session, if present)
- DELETE** `/sessions/{token}` Delete the session (if present) for the token sent

products : Data set operations [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

- POST** `/products` Create a new product
- GET** `/products` Get all products
- GET** `/products/{id}` Get product by id
- GET** `/products/{id}/details` Get the details for a product
- PATCH** `/products/{id}` Update a product
- DELETE** `/products/{id}` Delete the product (if present) for the product id sent

dev : Development & testing operations [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

- PATCH** `/dev/reset_db` Reset and reseed the database with test data

[BASE URL: `http://localhost:3000/docs/` , API VERSION: 1.0]

What's it not good for?

- Lots of data (presently)
- Binary data (??)

How to use Swagger

- on Github: [swagger-ui/dist](#)
- define your API with the “swagger spec”
- spec files are JSON files
- init a SwaggerUi object

Swagger Spec

- All categories defined in “index” JSON file
- Each category has a JSON file

Links

- github.com/wordnik/swagger-ui
- github.com/wordnik/swagger-spec
- github.com/DonVaughnDcs/swagger-demo