

servant

Webservice framework in Haskell

Архитектура REST

- клиент-серверная архитектура
- отсутствие состояния
- кэширование
- единообразие интерфейса
- промежуточные узлы (слои)

RESTful API

PUT `http://api.someshop.com/orders/42?status=delivered`

- API root
- путь к ресурсу
- параметры запроса

RESTful API

`http://api.someshop.com/orders`

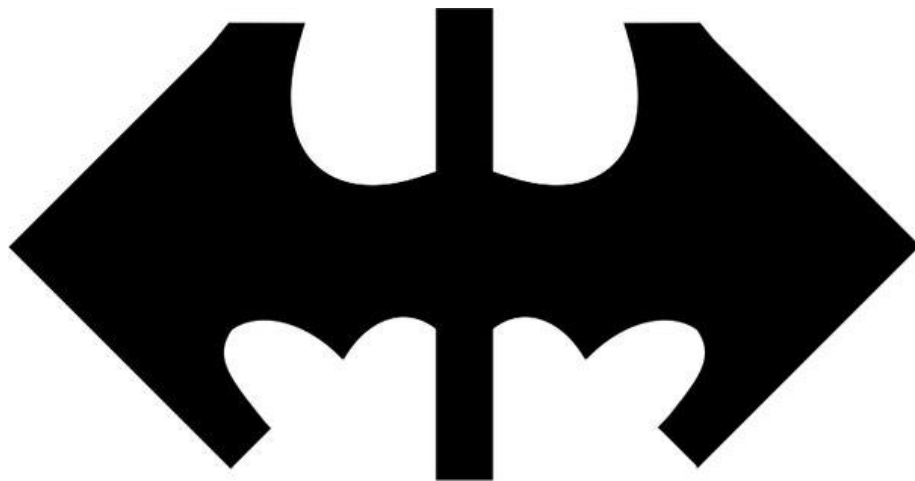
CRUD (**c**reate, **r**ead, **u**ppdate, **d**eleate)

- POST
- GET
- PUT
- DELETE

- /addNewItem
- /listAllDeliveredOrders
- /updateUserName
- /item/42/delete

...

servant



Bookshop API

- POST /books
- GET /books/:book_id
- PUT /books/:book_id?author=name&year=1984&in_stock=42...

Bookshop API

```
type BooksAPI
```

```
    = "books" :> ReqBody `[JSON] Book          :> Post `[JSON] BookId  
:<|> "books" :> Capture "book_id" BookId      :> Get `[JSON] Book  
:<|> "books" :> Capture "book_id" BookId      :> QueryParam "author" String  
          :> QueryParam "title" String        :> QueryParam "year" Int  
          :> QueryParam "in_stock" Int        :> Put `[JSON] Book
```


Bookshop API

- GET /books
 - POST /books
 - GET /books/:book_id
 - DELETE /books/:book_id
 - PUT /books/:book_id?author=name&year=1984&in_stock=42...
-
- GET /orders
 - POST /orders
 - GET /orders/:order_id
 - PUT /orders/:order_id?status=cancelled...

Bookshop API

```
type BookshopAPI
```

```
  = "books"    :> BooksAPI
```

```
  :<|> "orders" :> OrdersAPI
```

```
  :<|> "users"   :> UsersAPI
```

```
  :<|> "info"    :> InfoAPI
```

```
  ...
```

Bookshop API

```
type BooksAPI
```

```
    = Get `[JSON] [(BookId, Book)]
```

```
    :<|> ReqBody `[JSON] Book          :> Post `[JSON] BookId
```

```
    :<|> Capture "book_id" BookId      :> Get `[JSON] Book
```

```
    :<|> Capture "book_id" BookId      :> Delete `[JSON] ()
```

```
    :<|> Capture "book_id" BookId      :> QueryParam "author" String :>
```

```
        QueryParam "title" String :> QueryParam "year" Int          :>
```

```
        QueryParam "in_stock" Int :> Put `[JSON] Book
```

Time to code!



API type



- server
- client in Haskell
- client for JS, etc.
- documentation
- mock server

...

Контакты



`fierce_katie@xmpp.ru`



`vk.com/burtonist`



`github.com/fierce-katie`