Bobo

Bobo é um micro framework web, foi criado por Jim Fulton com notório trabalho reconhecido pelo desenvolvimento de outro framework, Zope, que é a base de CMS Plone, ERP5 e outros projetos Python de larga escala. Também é criador do banco de dados relacional ZODB (Zope Object Database) que oferece as propriedades ACID (atomicidade, consistência, isolamento e durabilidade). Jim Fulton esteve presente no evento de 2011 da Python Brasil e falou sobre o Zope.

Na época de seu lançamento em 1997 Bobo era pioneiro no conceito de mapeamento direto de URLs para objetos sem a necessidade de configurar rotas. Outra funcionalidade era o tratamento automático de query HTTP baseado na análise de assinaturas de métodos ou de funções usadas para tratar requisições.

Devido a evolução da linguagem Python Jim escreveu novamente o Bobo para oferecer suporte a WSGI e ao Python 3.

Exemplo para obter informações sobre parâmetros

O Bobo atualmente por meio da biblioteca six é capaz de fazer introspecção de função e identificar quais parâmetros são necessários na requisição.

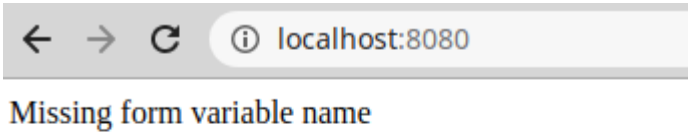
```
import bobo

@bobo.query('/')
def hello(name="world"):
     return 'Hello %s!' % name
```

O decorator bobo.query controla como a URL é mapeada para o objeto. Também controla como as funções são chamadas e os valores retornados convertido para respostas HTTP. Se é retornado string, assume que será HTML usado para construir uma resposta. No decorator é passado também o path da URL, no caso '/'.

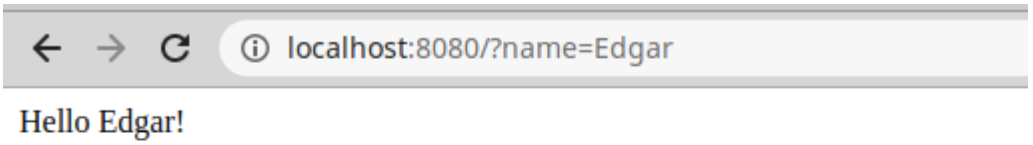
O decorator *query* entende que irá trabalhar com recursos que que retornem informações como por exemplo um formulário. Como pode ser observado na função o parâmetro *'name'* é necessário, fazendo a omissão dele como no código abaixo vai ocasionar erro apontando sua ausência.

```
@bobo.query('/')
def hello(name):
     return 'Hello %s!' % name
```



Como a função suporta form data se acessar o link a seguir irá ser exibido o conteúdo pois foi satisfeito o parâmetro.

http://localhost:8080/?name=Edgar



Podemos observar que com o uso de decoradores podemos implementar novas funcionalidades e modificar o comportamento da função sem alterações diretas.

Referências

Documentação: https://bobo.readthedocs.io/en/latest/#getting-started
Repositório: https://pypi.org/project/bobo/
Livro: Python Fluente: Programação Clara, Concisa e Eficaz, Luciano Ramalho