Fatec
Rio Claro

Linguagem de Programação II

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

FastAPI

FastAPI

Fatec
Rio Claro

O FastAPI foi anunciado em 2018 por Sebastián Ramírez. É mais moderno em muitos sentidos do que a maioria dos frameworks web Python, aproveitando os recursos adicionados ao Python 3 nos últimos anos.

Como qualquer framework web, o FastAPI ajuda você a construir aplicações web. Cada framework é projetado para facilitar algumas operações, por meio de recursos e padrões.

FastAPI é voltado para o desenvolvimento de APIs web, embora você também possa usá-lo para aplicações tradicionais de conteúdo web.

A Web Moderna

Web Moderna

Fatec
Rio Claro

Era uma vez a web pequena e simples. Os desenvolvedores se divertiam muito juntando chamadas PHP, HTML e MySQL em arquivos únicos e dizendo orgulhosamente a todos para visitarem seus sites. Mas a web cresceu com o tempo para zilhões, ou melhor, quilhões de páginas - e o playground inicial se tornou um metaverso de parques temáticos.

Serviços e APIs

Fatec
Rio Claro

A web é uma ótima rede de conexões. Embora ainda haja muita atividade no que diz respeito ao conteúdo — HTML, JavaScript, imagens e assim por diante —, há uma ênfase crescente nas interfaces de programação de aplicativos (APIs) que conectam as coisas.

Comumente, um serviço web lida com acesso a bancos de dados de baixo nível e lógica de negócios de nível médio (frequentemente agrupados como backend), enquanto JavaScript ou aplicativos móveis fornecem um frontend de alto nível (interface de usuário interativa).

Esses mundos de vanguarda tornaram-se mais complexos e divergentes, geralmente exigindo que os desenvolvedores se especializem em um ou outro. É mais difícil ser um desenvolvedor full stack do que costumava ser.

Serviços e APIs

Fatec
Rio Claro

Vários métodos de API foram desenvolvidos à medida que a tecnologia evoluiu de máquinas isoladas para sistemas multitarefas e servidores em rede.

Antes das redes, uma API geralmente significava uma conexão muito próxima, como uma chamada de função para uma biblioteca na mesma linguagem que sua aplicação — digamos, calcular uma raiz quadrada em uma biblioteca matemática.

Chamadas de procedimento remoto (RPCs) foram inventadas para chamar funções em outros processos, na mesma máquina ou em outras, como se estivessem na aplicação que as chama. Um exemplo popular atual é o gRPC.

Serviços e APIs

Fatec
Rio Claro

O sistema de mensagens envia pequenos blocos de dados em pipelines entre processos.

As mensagens podem ser comandos semelhantes a verbos ou podem apenas indicar eventos de interesse semelhantes a substantivos.

As soluções de mensagens populares atuais, que variam amplamente de kits de ferramentas a servidores completos, incluem Apache Kafka, RabbitMQ, NATS e ZeroMQ.

Serviços e APIs

Fatec
Rio Claro

A comunicação pode seguir diferentes padrões:

Solicitação-resposta: Como um navegador web chamando um servidor web.

Publicar-assinar ou pub-sub: Um publicador emite mensagens, e os assinantes agem em cada uma delas de acordo com alguns dados da mensagem, como um assunto.

Filas: Como pub-sub, mas apenas um de um conjunto de assinantes pega a mensagem e age sobre ela.

HTTP

Fatec
Rio Claro

Berners-Lee propôs três componentes para sua World Wide Web:

HTML: Uma linguagem para exibição de dados

HTTP: Um protocolo cliente-servidor

URLs: Um esquema de endereçamento para recursos da web

Embora pareçam óbvios em retrospecto, acabaram se revelando uma combinação extremamente útil.

REST(ful)

Fatec
Rio Claro

Um capítulo da tese de doutorado de Roy Fielding definiu a Transferência Representacional de Estado (REST) — um estilo arquitetônico para uso em HTTP. Embora frequentemente referenciada, tem sido amplamente mal compreendida.

Uma adaptação, mais ou menos compartilhada, evoluiu e domina a web moderna. Ela se chama RESTful, com as seguintes características:

- Usa HTTP e protocolo cliente-servidor
- Sem estado (cada conexão é independente)
- Cacheável
- Baseado em recursos

Um recurso são dados que você pode distinguir e nos quais pode executar operações. Um serviço web fornece um ponto de extremidade - uma URL e um verbo HTTP (ação) distintos - para cada recurso que deseja expor.

Um ponto de extremidade também é chamado de rota, porque ele direciona a URL para uma função.

REST(ful)

Fatec
Rio Claro

Usuários de banco de dados estão familiarizados com a sigla CRUD para procedimentos: criar, ler, atualizar, excluir. Os verbos HTTP são bem CRUD:

POST Criar (escrever)

PUT Modificar completamente (substituir)

PATCH Modificar parcialmente (atualizar)

GET Obter (ler, recuperar)

DELETE Excluir

REST(ful)

Fatec
Rio Claro

Um cliente envia uma solicitação a um endpoint RESTful com dados em uma das seguintes áreas de uma mensagem HTTP:

Cabeçalhos (*Headers*)

A sequência de caracteres da URL

Parâmetros da consulta (*Query parameters*)

Valores do corpo (*Body values*)

Status Code

Fatec
Rio Claro

Por sua vez, uma resposta HTTP retorna o seguinte:

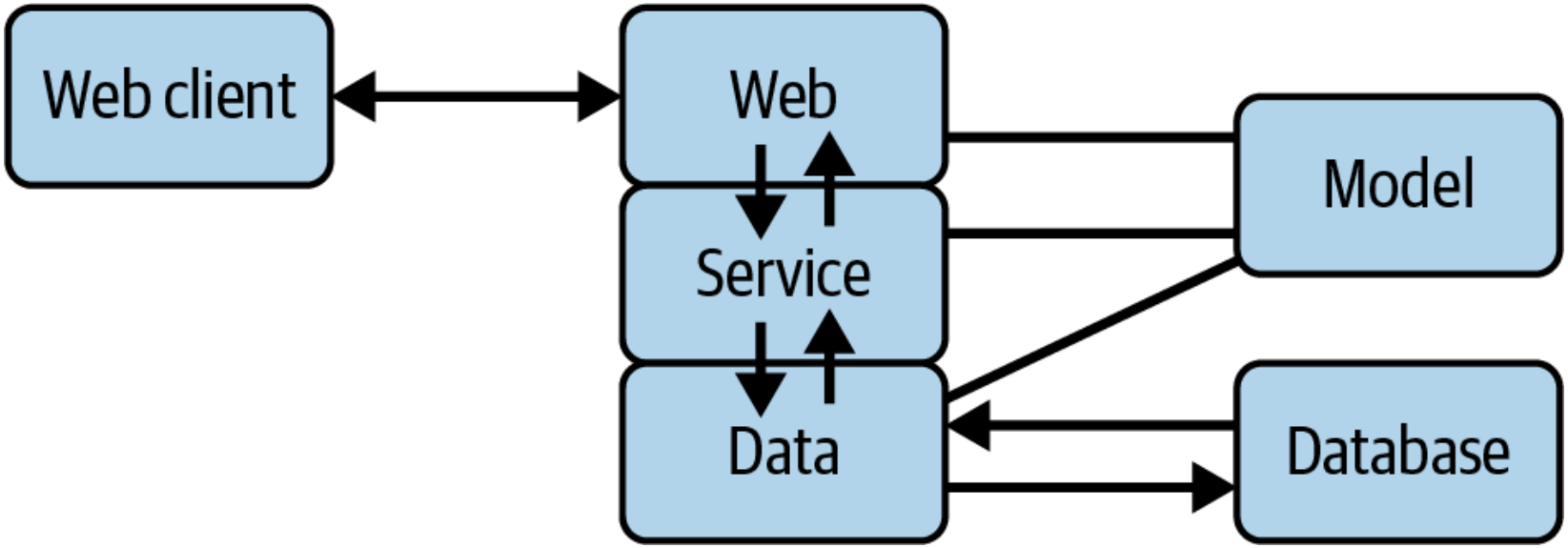
Um código de status inteiro indicando o seguinte:

100s	Informações, continue
200s	Sucesso
300s	Redirecionamento
400s	Erro do cliente
500s	Erro do servidor

Camadas

Fatec
Rio Claro

Para gerenciar tamanho e complexidade, muitas aplicações utilizam há muito tempo o chamado modelo de três camadas.



Camadas

Fatec
Rio Claro

Web Camada de entrada/saída sobre HTTP, que reúne solicitações de clientes, chama a Camada de Serviço e retorna respostas

Serviço A lógica de negócios, que chama a camada de Dados quando necessário

Dados Acesso a repositórios de dados e outros serviços

Modelo Definições de dados compartilhadas por todas as camadas

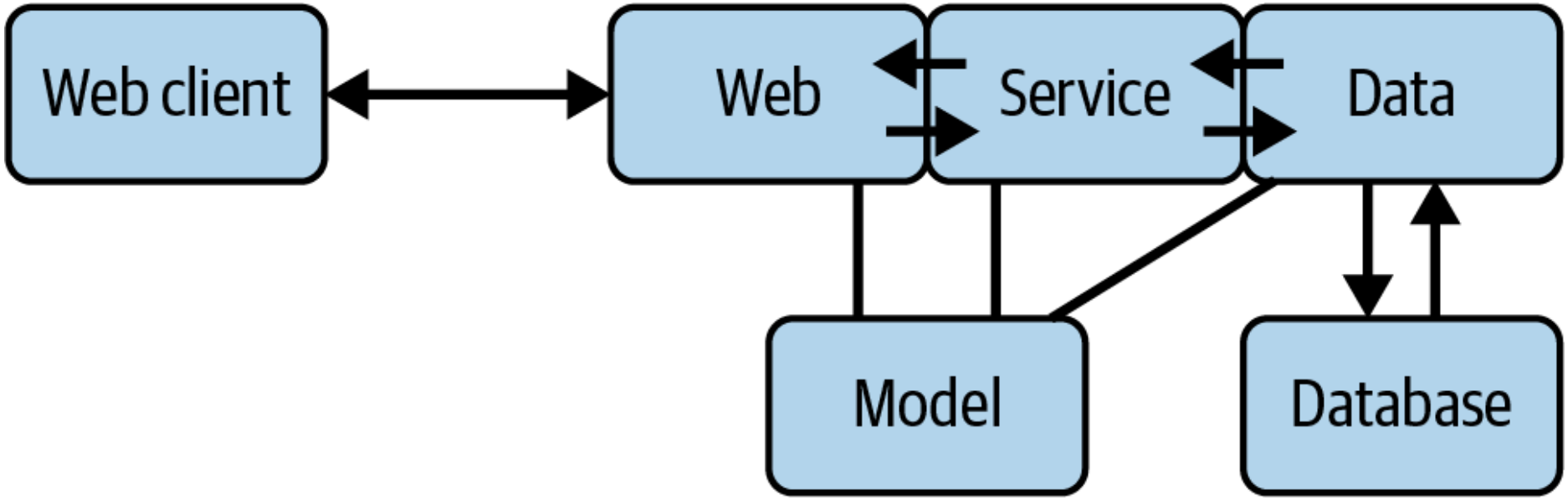
Cliente Web Navegador Web ou outro software do lado do cliente HTTP

Banco de Dados O repositório de dados, geralmente um servidor SQL ou NoSQL

Camadas

Fatec
Rio Claro

As camadas se comunicam entre si por meio de APIs. Elas podem ser simples chamadas de função para módulos Python separados, mas podem acessar código externo por qualquer método.



Camadas

Fatec
Rio Claro

A separação e a ocultação de informações são tratadas pelos módulos.

A **camada Web** é aquela que os usuários veem, por meio de aplicativos clientes e APIs. Geralmente, estamos falando de uma interface web RESTful, com URLs e solicitações e respostas codificadas em JSON.

A **camada de Serviço** contém os detalhes reais de tudo o que este site fornece. Essa camada se parece essencialmente com uma biblioteca. Ela importa módulos de Dados para acessar bancos de dados e serviços externos, mas não deve conhecer os detalhes.

A **camada de Dados** fornece à camada de Serviço acesso aos dados, por meio de arquivos ou chamadas de clientes para outros serviços. Camadas de Dados alternativas também podem existir, comunicando-se com uma única camada de Serviço.

FastAPI

FastAPI

Fatec
Rio Claro

Instale os pacotes básicos do Python que usaremos:

O framework FastAPI: pip install fastapi

O servidor web Uvicorn: pip install uvicorn

O cliente web de texto HTTPie: pip install httpie

O pacote do cliente web síncrono Requests: pip install requests

O pacote do cliente web síncrono/assíncrono HTTPX: pip install httpx

Prática

Fatec
Rio Claro

Observe os códigos: **hello.py**

Execute o comando: **uvicorn hello:app --reload**

Em outro terminal:

http -b localhost:8000/hi

http -v localhost:8000/hi

```
import requests
r = requests.get("http://localhost:8000/hi")
r.json()
```

Prática

Fatec
Rio Claro

Observe os códigos: **hello2.py**

Em outro terminal:

http -b localhost:8000/hi

http -v localhost:8000/hi

```
import requests
r = requests.get("http://localhost:8000/hi")
r.json()
```

Prática

Fatec
Rio Claro

Observe os códigos: **hello3.py**

Em outro terminal:

http -b localhost:8000/hi/Fatec

http -v localhost:8000/hi/Fatec

```
import requests
r = requests.get("http://localhost:8000/hi/Fatec")
r.json()
```

Neste caso, a string “Fatec” é passado como parte da URL.

A resposta em cada caso é a string JSON (com aspas simples ou duplas, dependendo de qual cliente de teste que você usou)

Já os parâmetros de consulta (*Query Parameters*) são strings

nome=valor

depois do ? em uma URL, separado por &

Prática

Fatec
Rio Claro

Observe os códigos: **hello4.py**

Em outro terminal:

http -b localhost:8000/hi?who=Fatec

http -v localhost:8000/hi?who=Fatec

```
import requests
r = requests.get("http://localhost:8000/hi?who=Fatec")
r.json()
```

FastAPI

Fatec
Rio Claro

Podemos fornecer parâmetros de caminho ou consulta para um endpoint GET, mas não valores do corpo da solicitação.

Em HTTP, GET deve ser idempotente — um termo computacional para "fazer a mesma pergunta e obter a mesma resposta".

O GET do HTTP deve retornar apenas informações.

O corpo da solicitação é usado para enviar informações ao servidor ao criar (POST) ou atualizar (PUT ou PATCH).

No próximo exemplo, Body(embed=True) é necessário para informar ao FastAPI que, desta vez, obtemos o valor de who do corpo da solicitação em formato JSON.

Prática

Fatec
Rio Claro

Observe os códigos: **hello5.py**

Em outro terminal:

http -b localhost:8000/hi who=Fatec

http -v localhost:8000/hi who=Fatec

```
import requests
r = requests.post("http://localhost:8000/hi", json={"who": "Fatec"})
r.json()
```

FastAPI

Fatec
Rio Claro

Finalmente, vamos tentar passar o argumento de saudação como um cabeçalho HTTP.

Vamos testar isso apenas com HTTPie no hello6.

Ele usa nome:valor para especificar um cabeçalho HTTP.

Prática

Fatec
Rio Claro

Observe os códigos: **hello6.py**

Em outro terminal:

```
import requests
r = requests.post("http://localhost:8000/hi", headers={"who": "Fatec"})
r.json()
r = requests.post("http://localhost:8000/agent")
r.json()
```

Fatec
Rio Claro

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

Bibliografia

Myers, Jason; Copeland, Rick. Essential SQLAlchemy: Mapping Python to Databases. 2. ed. Sebastopol, CA: O'Reilly Media, 2015. 208 p. ISBN 978-1491916469.

Bibliografia

VORON, François. Building Data Science Applications with FastAPI: Harness the Power of Python, FastAPI, and Modern Tools to Build Production-Ready Data Science Applications. 2. ed. Birmingham: Packt Publishing, 2023.