

Building a REST API with Python and Flask

ADDING AUTHENTICATION TO OUR API

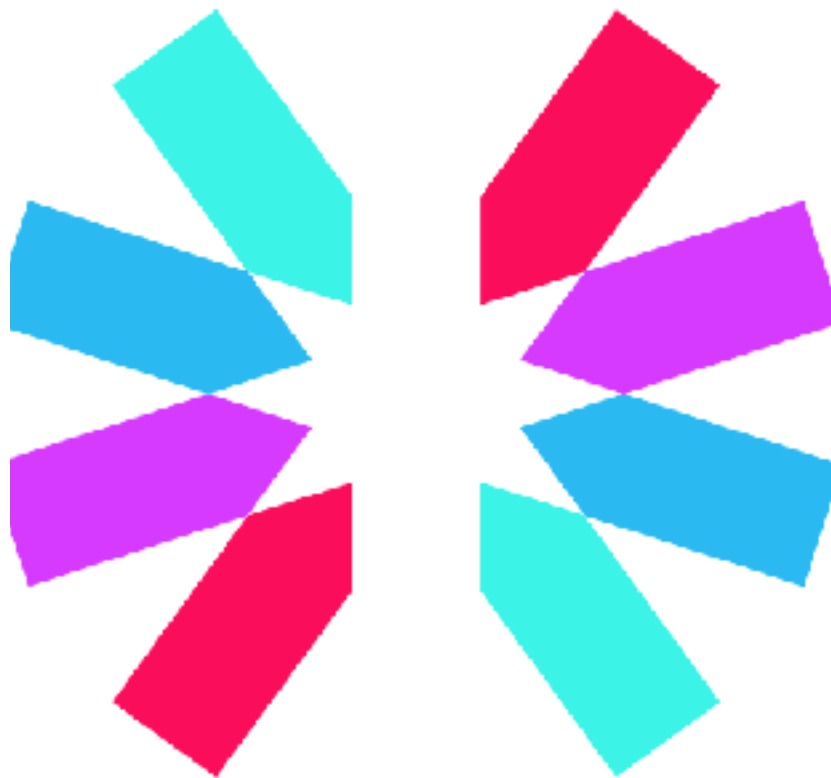


SANJAY RAI
PLURALSIGHT AUTHOR

Clients Sending Tokens for API Access



JSON Web Tokens (JWT)



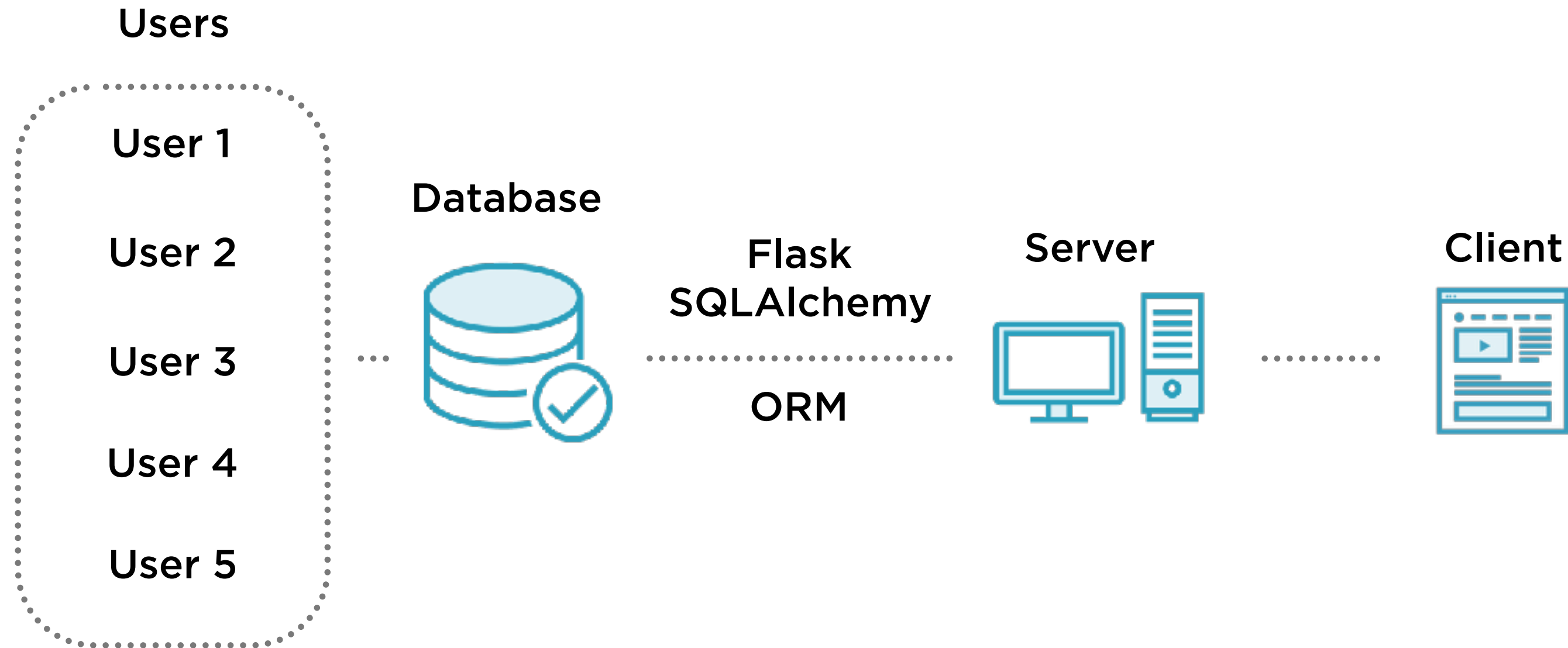
```
// Header
```

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

```
// Payload
```

```
{  
  "id": 1234567890  
  "name": "John Doe"  
  "issued_at": 1516239022  
  "expiration_date": 1516939022  
}
```

How Our Database Will Be Setup



Using Decorators to Add Authentication

```
@app.route('/books', methods=['POST'])
```

```
@token_required
```

```
def add_book():
```

```
@app.route('/books/<int:isbn>', methods=['PUT'])
```

```
@token_required
```

```
def replace_book():
```

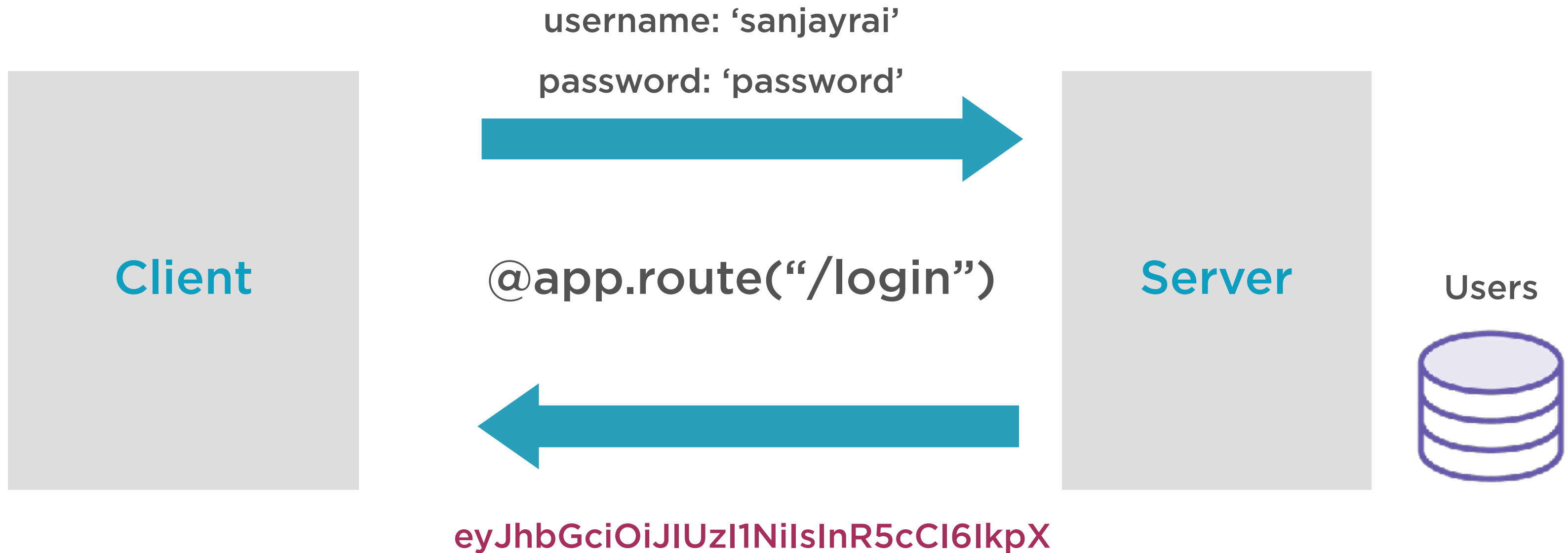
```
@app.route('/books/<int:isbn>', methods=['PATCH'])
```

```
@token_required
```

```
def update_book():
```

Creating a Database for Users Who Are Allowed API Access

Giving Tokens Only to Authenticated Users



On production code use an authentication library.

Review

Created Our First 'Hello World' Flask App

Built Routes for the Following HTTP Methods

- GET
- POST
- PUT
- PATCH
- DELETE

Migrated All of Our Data to a Database

Next Steps



Flask-RESTful

- Extension to organize large projects
- <https://flask-restful.readthedocs.io/en/latest/>

Flask JWT Extended

- Used if you have unique JWT use cases in your project
- <https://flask-restful.readthedocs.io/en/latest/>

Flask Official Documentation

- <http://flask.pocoo.org/docs/>