

# Creating the Sample Application

---



**Justin Menga**

FULL STACK TECHNOLOGIST

@jmenga [pseudo.co.de](https://pseudo.co.de)

# Introduction

## **Python Application**

- Django REST Framework Web Service
- Based upon <http://todobackend.com>

## **Application Tests**

- Unit and Integration Tests
- Acceptance Tests

## **Deployment Tasks**

- e.g. Application configuration

# Why a Python Sample Application?

## Easy to Develop

Develop from text editor  
Functional within minutes  
Test driven

## Django REST

RESTful Services  
Model driven  
Web Interface

## Difficult to Deploy

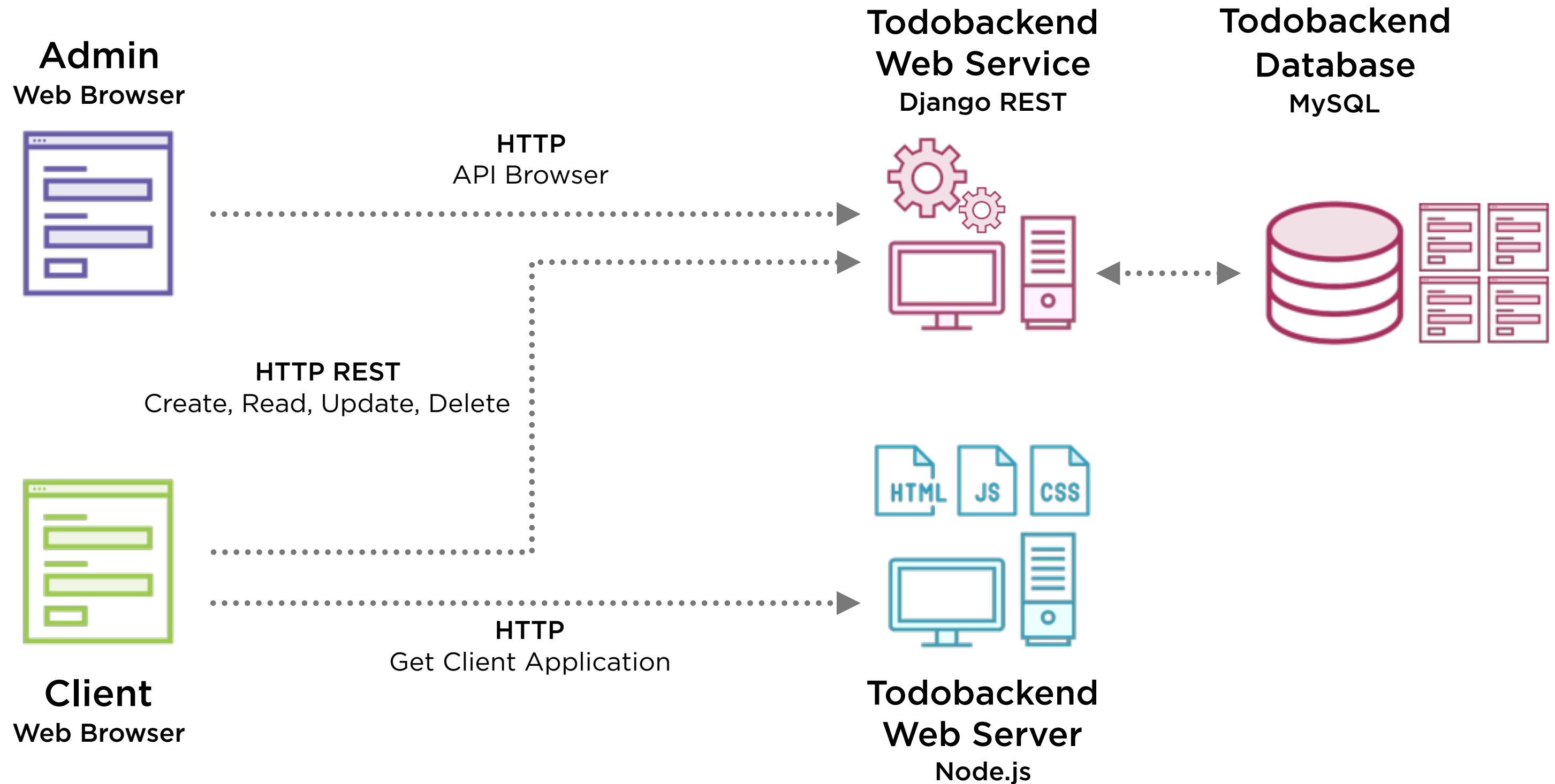
Dependency Management  
Packaging  
Virtual Environments

# Demo

## **Application Walkthrough**

- Architecture
- Front End Interface
- Back End Interface

# Application Architecture



# Demo

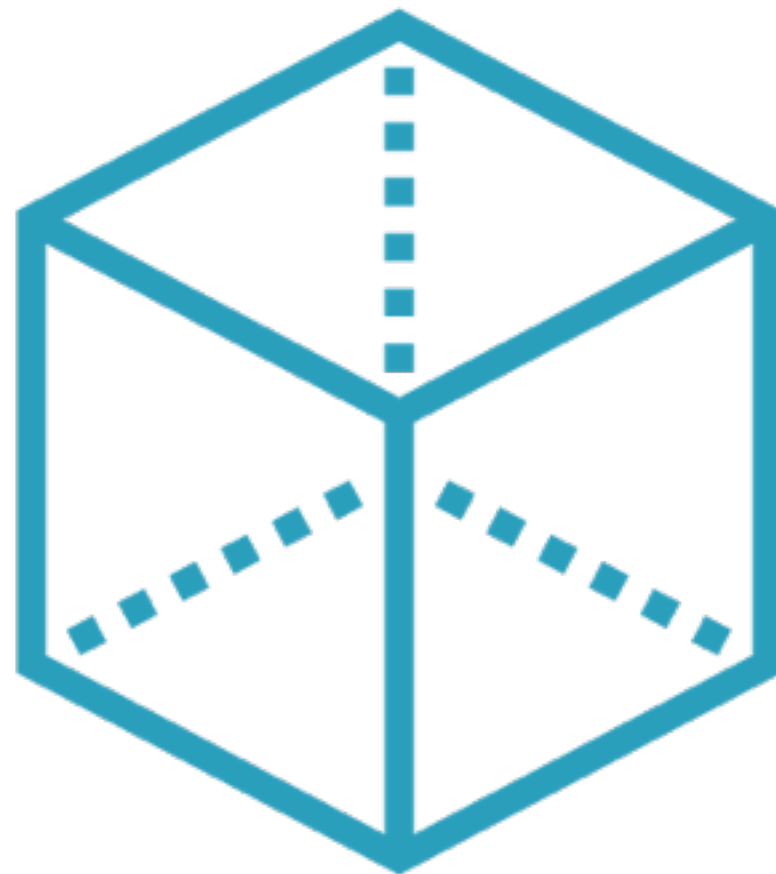
## Creating the Todobackend Web Service

- Initial setup
- Creating models
- Creating serializers
- Creating views
- Configuring routing
- Test driving the application

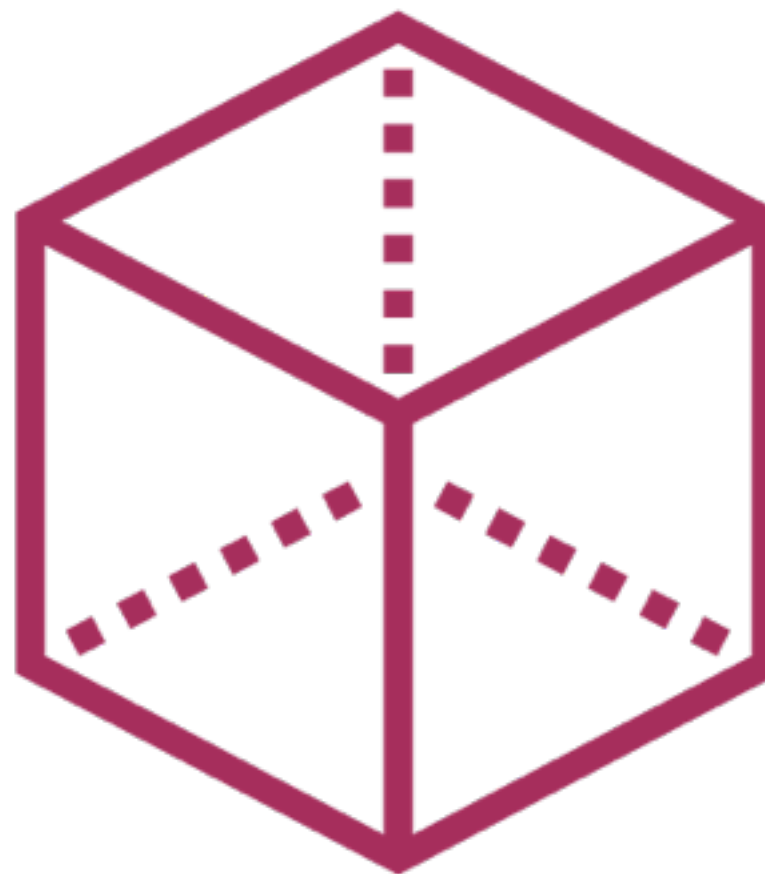
# Initial Setup

---

# Virtual Environments



**Virtual Environment**  
Python 3



**Virtual Environment**  
Python 2.7



**Virtual Environment**  
Python 2.7



# Creating Models

---

# Creating Serializers

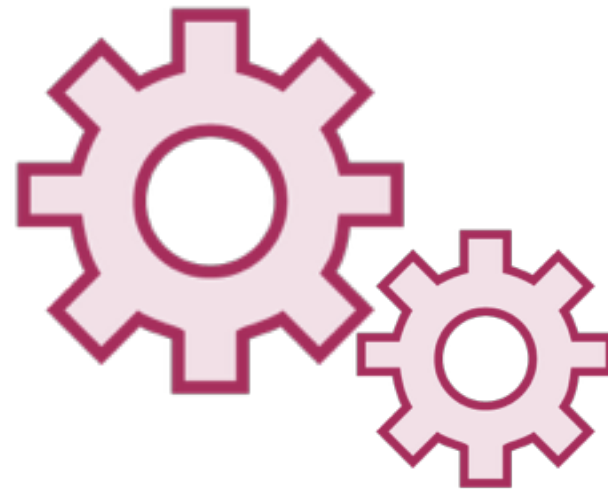
---

# Serialization

**Wire  
Representation**  
e.g. JSON



**De-serialization**



**Serialization**

**Model  
Representation**  
e.g. Database Record



# Creating Views

---

# Views

**HTTP Request**  
Method: POST (Create)

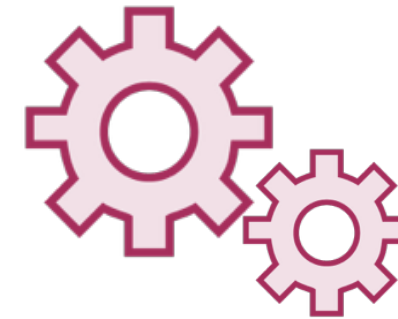


**View Engine**



**Detail View**  
(Single Item)

Serialization  
Deserialization



Model



**List View**  
(Multiple Items)

Serialization  
Deserialization



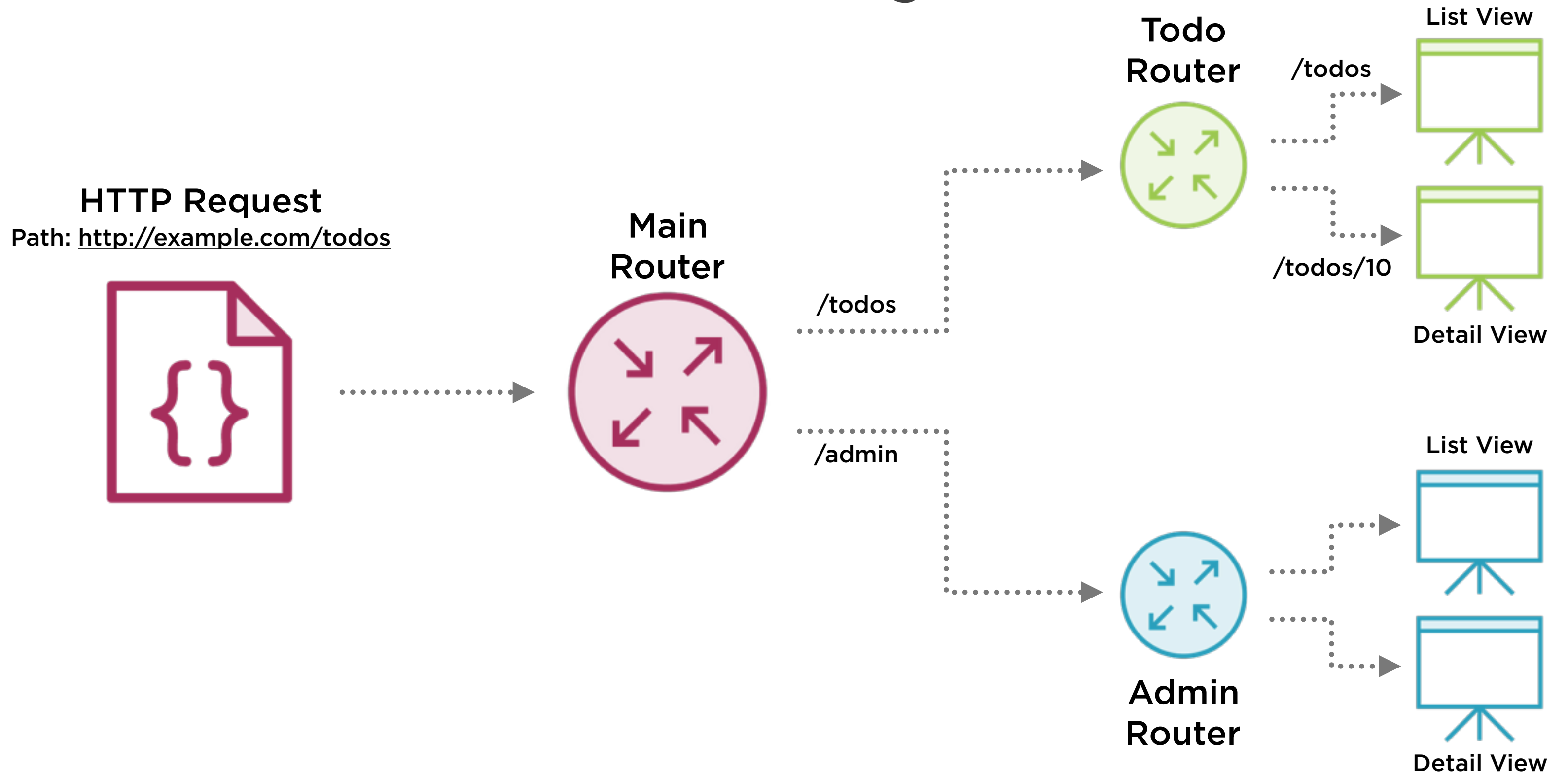
Model



# Configuring Routing

---

# Routing



# Test Driving the Application

---



# Demo

## **Unit and Integration Testing**

- Creating tests
- Refactoring settings
- Configuring integration tests
- Improving test output

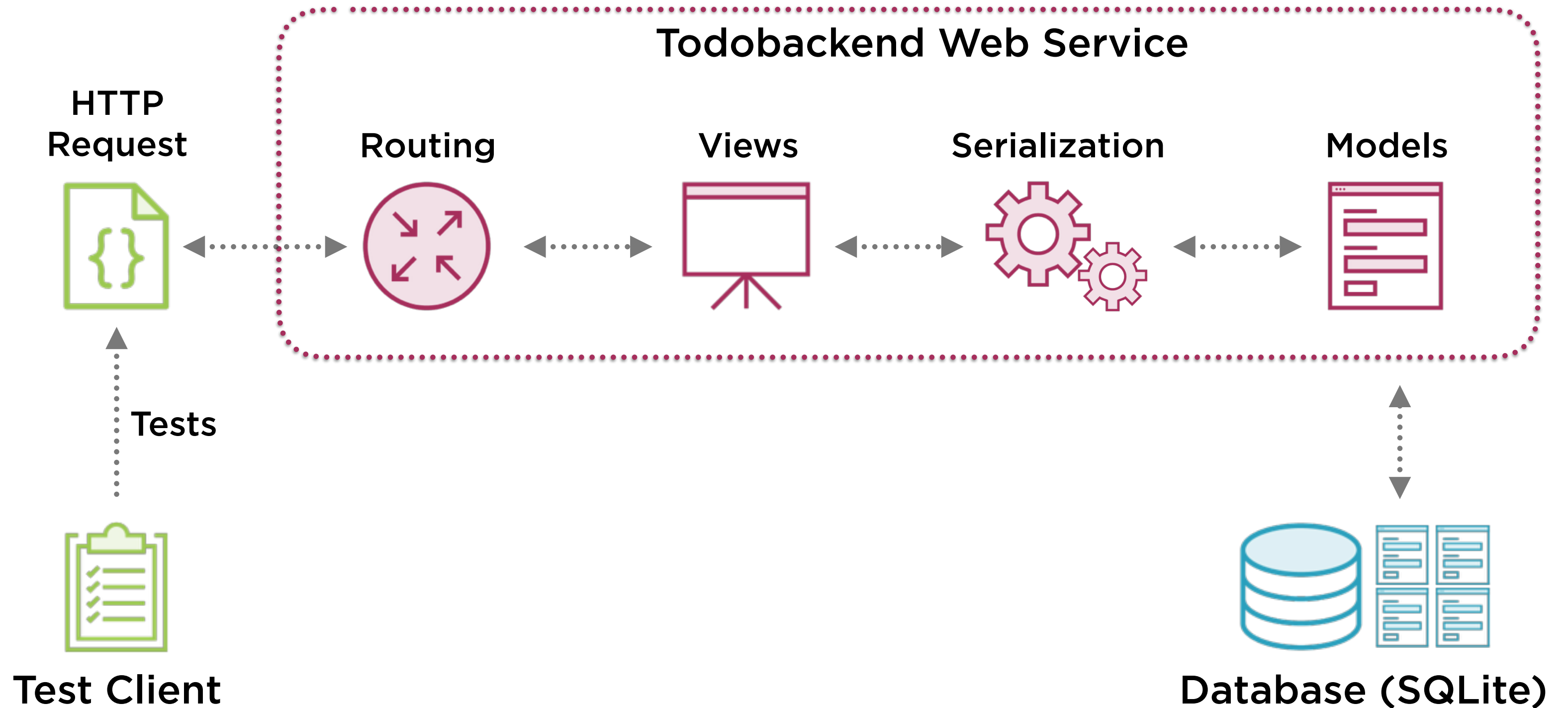
# Creating Tests

---

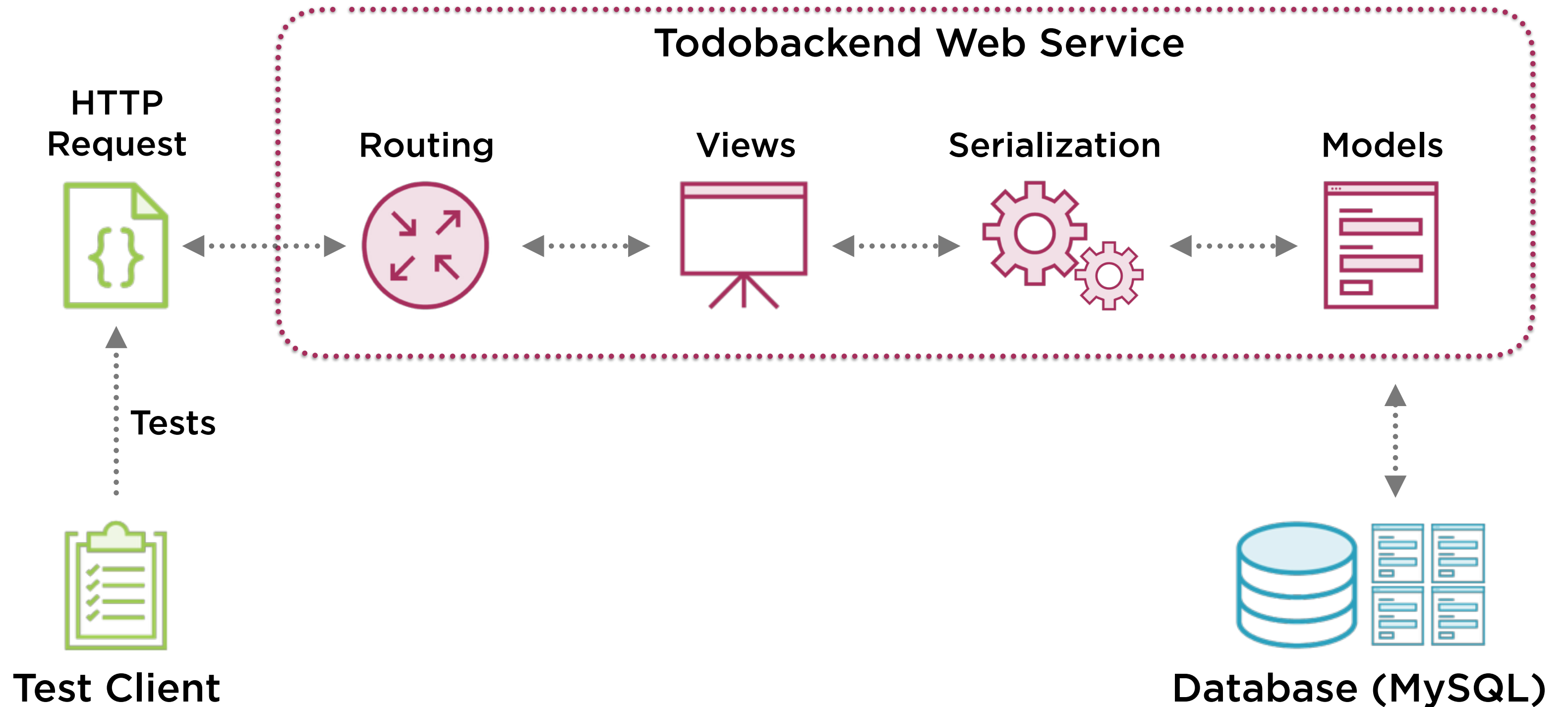
# Unit vs Integration Tests

---

# Integration Testing



# Integration Testing



# Refactoring Settings

---

# Configuration Settings

## Release Settings



Engine: MySQL  
Host: db  
DB: todobackend  
Password: \*\*\*\*\*  
Debug: True



## Test Settings



Engine: SQLite  
Host: localhost  
Debug: True



## QA Environment Settings



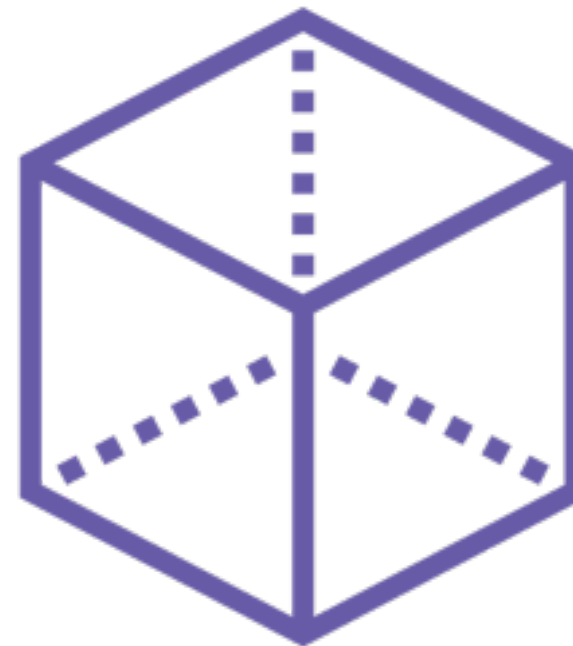
Engine: MySQL  
Host: qasql01  
DB: todobackend  
Password: \*\*\*\*\*  
Debug: False



## Prod Environment Settings

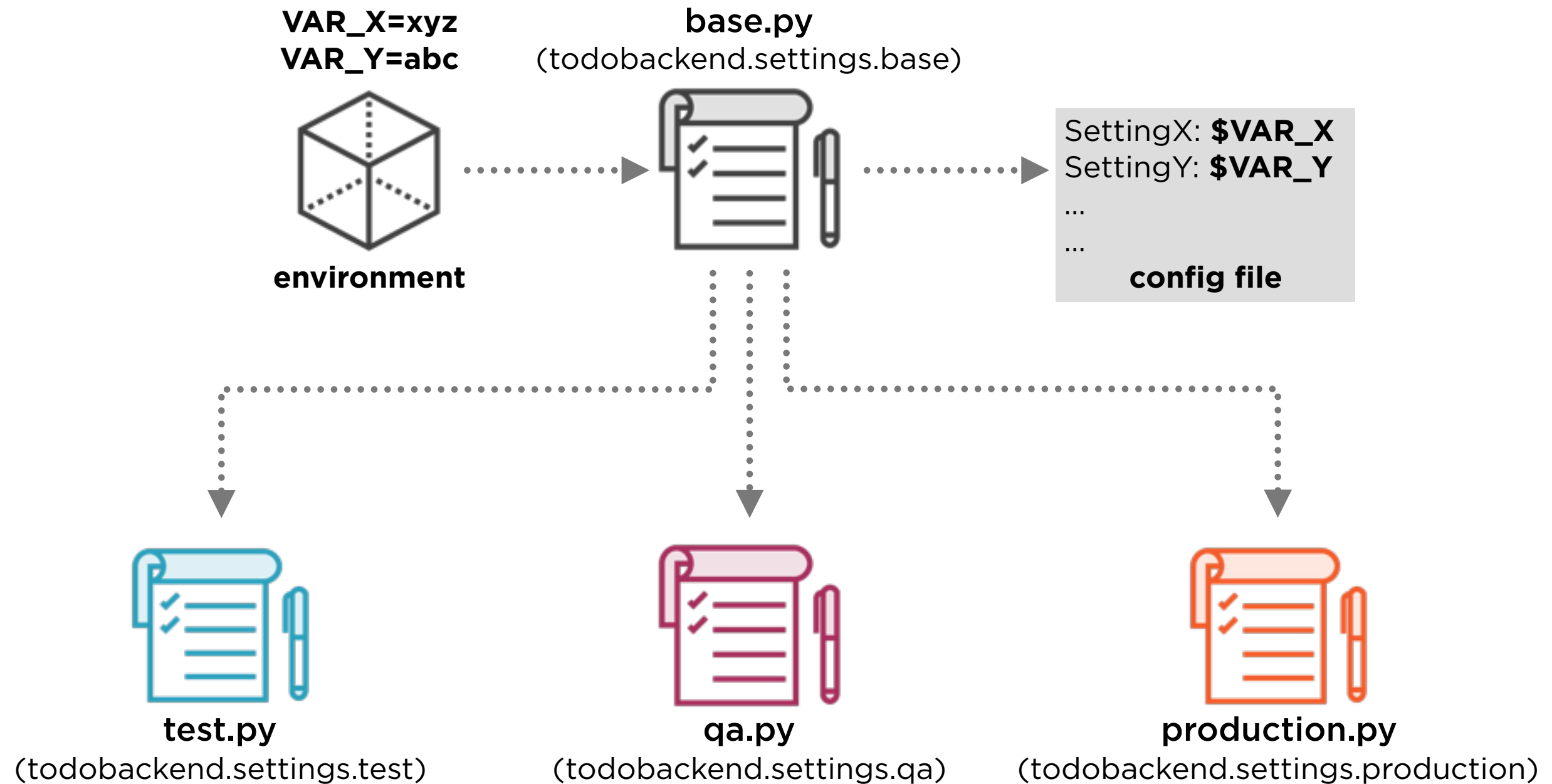


Engine: MySQL  
Host: proddb01  
DB: todobackend  
Password: \*\*\*\*\*  
Debug: False



## Application

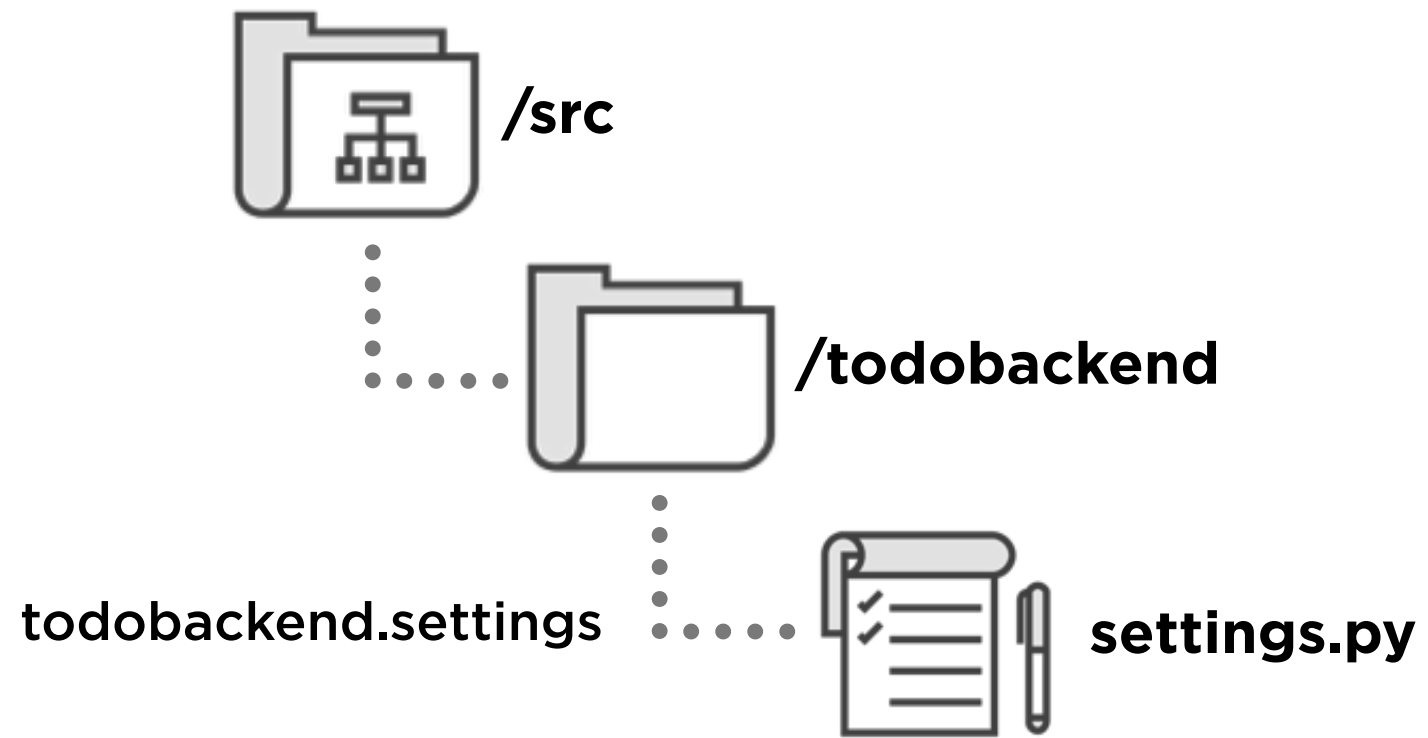
# Configuration Settings



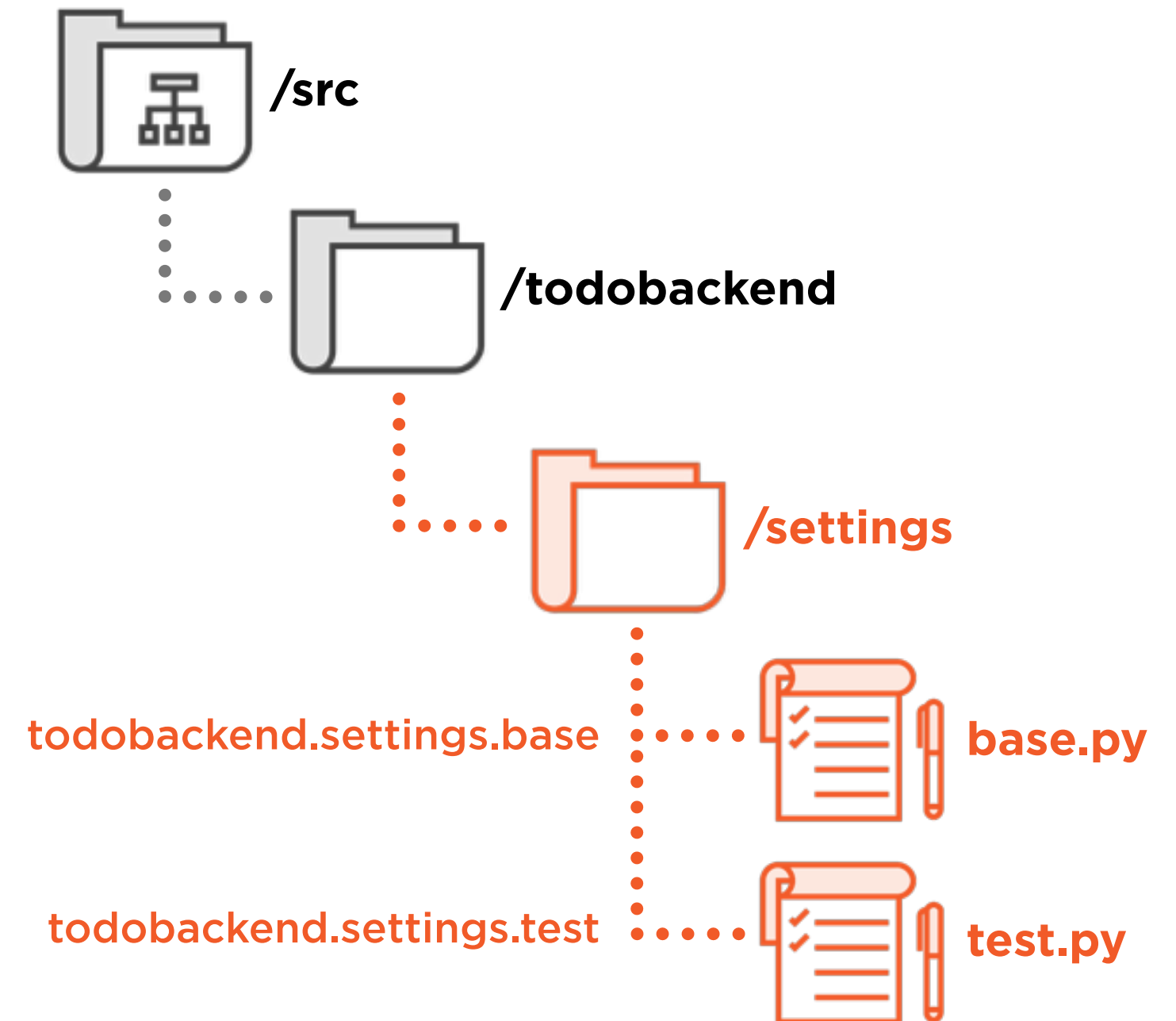


# Configuration Settings

## Default Settings Layout



## New Settings Layout



# Configuring Integration Tests

---

# Installing MySQL

---

# Improving Test Output

---

# Demo

## **Acceptance Testing**

- Initial setup
- Writing acceptance tests
- Running acceptance tests

# Initial Setup

---

# Creating Acceptance Tests

---

# Running Acceptance Tests

---



# Demo

## **Testing the user interface**

- Installing the Todobackend client
- Verifying the user interface

# Installing the Todobackend Client

---

# Verifying the User Interface

---

# Summary

## **Sample Application**

- How to configure
- How to run
- How to test

## **Testing**

- Unit and Integration Tests
- Acceptance Tests using Node.js
- Manual Testing