

Passo 1: Configurar o Banco de Dados

Primeiro, você precisa configurar um banco de dados para armazenar informações sobre as chaves. Vamos usar o SQLite para fins de demonstração. Certifique-se de instalar o pacote `flask_sqlalchemy` usando `pip install flask_sqlalchemy`.

```
# app/database.py
from flask_sqlalchemy import SQLAlchemy
db = SQLAlchemy()
```

Configure o aplicativo Flask para usar o banco de dados.

```
# app/__init__.py
from flask import Flask
from app.controllers.chave_controller import chave_bp
from app.database import db

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///chaves.db' # Use SQLite como banco de dados
db.init_app(app)
app.register_blueprint(chave_bp)

if __name__ == '__main__':
    app.run()
```

Passo 2: Definir o Modelo do Banco de Dados

Dentro do diretório `models`, crie um arquivo `chave.py` para definir o modelo do banco de dados.

```
# app/models/chave.py
from app.database import db

class Chave(db.Model):
    __tablename__ = 'chaves'

    chave_id = db.Column(db.Integer, primary_key=True)
    descricao = db.Column(db.String(255))
    status = db.Column(db.String(20))

    def __init__(self, descricao, status):
        self.descricao = descricao
        self.status = status

    def to_dict(self):
        return {'chave_id': self.chave_id, 'descricao': self.descricao, 'status': self.status}
```

Passo 3: Modificar o Controlador

Dentro do diretório `controllers`, modifique o arquivo `chave_controller.py` para buscar as chaves no banco de dados.

```
# app/controllers/chave_controller.py
from flask import Blueprint, jsonify
from app.models.chave import Chave
from app.database import db

chave_bp = Blueprint('chave', __name__, url_prefix='/api')

@chave_bp.route('/chaves_disponiveis', methods=['GET'])
def listar_chaves_disponiveis():
    chaves = Chave.query.filter_by(status='disponível').all()
    chaves_disponiveis = [chave.to_dict() for chave in chaves]
    return jsonify({"chaves_disponiveis": chaves_disponiveis})
```

Passo 4: Criar o Banco de Dados e Preencher Dados Iniciais

Agora que você configurou o banco de dados, crie as tabelas e preencha alguns dados iniciais.

```
cd project/app
python
```

Dentro do shell Python:

```
from app import db
db.create_all()

from app.models.chave import Chave
chave_a = Chave("Chave A", "disponível")
chave_b = Chave("Chave B", "disponível")
db.session.add(chave_a)
db.session.add(chave_b)
db.session.commit()
exit()
```

Passo 5: Executar o Aplicativo

Agora, você pode executar o aplicativo Flask seguindo os mesmos passos mencionados anteriormente:

```
cd project/app
python __init__.py
```

E então, execute o script do frontend:

```
cd project/frontend python main.py
```