

البيانات الابية

البيانات الابية هي مجموعات من المعلومات التي تستخدم في تطبيقات البرمجيات. تمثل البيانات الابية المعرفة ببيانات المدخلات والخرجات، وبيانات التحكم والسيطرة، وبيانات المراقبة والرسائل.

1. الاعتماديات إضافة

البيانات الابية يمكن إضافتها إلى مشروع فـ `pom.xml` بحسب النسبة. إذا كنت إذاً في حاجة إلى إضافة علائقية مثل `spring-boot-starter-data-jpa` أو `spring-boot-starter-web`، فيجب أن ت 添加 إلى ملف `pom.xml`.

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <scope>runtime</scope>
</dependency>
```

2. البيانات قاعدة تهيئة

البيانات قاعدة تهيئة يمكن إضافتها إلى ملف `application.properties`. إذا كنت إذاً في حاجة إلى إعدادات بـ `application.properties`، فيجب أن ت `application.properties` في المثالي.

```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=password
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
```

3. كيان إنشاء

التحوييل: لتحديد `application.properties` تعليقات استخدم البيانات. قاعدة جدول مع تتوافق كيان فئة بـ `application.properties`.

```

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private String email;

    // Getters and setters
}

```

4. مستودع واجهة إنشاء.

لكل كيان **User** عمليات الواجهة هذه ستتوفر JpaRepository. من تمتد مستودع واجهة بإنشاء قم:

```

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {
    //
}

```

5. خدمة في المستودع استخدام.

البيانات: قاعدة عمليات لإجراء الخدمة فئة في المستودع بـ إدخال قم

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;

@Service
public class UserService {
    @Autowired
    private UserRepository userRepository;

    public List<User> getAllUsers() {

```

```

    return userRepository.findAll();
}

public User getUserById(Long id) {
    return userRepository.findById(id).orElse(null);
}

public User saveUser(User user) {
    return userRepository.save(user);
}

public void deleteUser(Long id) {
    userRepository.deleteById(id);
}

```

6. إنشاء الـ REST API

الكيان مع للتفعال النهاية نساطة تقدم `User` بإنشاء قم:

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/users")
public class UserController {
    @Autowired
    private UserService userService;

    @GetMapping
    public List<User> getAllUsers() {
        return userService.getAllUsers();
    }

    @GetMapping("/{id}")
    public User getUserById(@PathVariable Long id) {
        return userService.getUserById(id);
    }
}

```

```

@PostMapping
public User createUser(@RequestBody User user) {
    return userService.saveUser(user);
}

@DeleteMapping("/{id}")
public void deleteUser(@PathVariable Long id) {
    userService.deleteUser(id);
}

```

٧. تطبيق تشغيل الات

User الکیان علی عملیات تنفيذی علی آن قادرًا تكون أن يجب بك. الخاص `@PostMapping` تطبيق بتشغيل قم `@DeleteMapping` `{id}` `@PathVariable` `Long id`.

أكبر بشكل توسيعه يمكنك . مشروع في بيانت لمصر در مستودع لاستخدام أساسية بنية الإعداد هذا يوفر تعقيداً. أكثر أعمال منطق وتطبيق واستثناءات، معالجة مخصوصة، أسئلة إضافة خل ال من