

+ New chat

OTP Code Implementati

application.properties and app

Using WinSCP with Spring Boot

Clear conversations

Upgrade to Plus

Dark mode

Updates & FAQ

Log out

OTP verify karne ke liye

S

ky apn aache se pura code lkh sakte he otp ko implement karne ke liye spring boot project me

Janur, mai aapko Spring Boot project me OTP implementation karne ke liye sample code provide kar sakta hu.

Step 1: Library aur Dependency Add Karna
Sabse pehle, hum pom.xml file me dependency add karenge, jisse hum TOTP generate aur verify kar sake.

xml

<dependency>
 <groupId>com.warrenstrange</groupId>
 <artifactId>googleauth</artifactId>
 <version>1.2.1</version>
</dependency>

Step 2: OTP Generate Karna
OTP generate karne ke liye, hum "GoogleAuthenticator" class ka use karenge. Ye class OTP generate aur verify karne ke liye methods provide karta hai.

java

import com.warrenstrange.googleauth.GoogleAuthenticator;
import com.warrenstrange.googleauth.GoogleAuthenticatorConfig;
import com.warrenstrange.googleauth.GoogleAuthenticatorKey;

public class OtpService {
 public String generateSecretKey() {
 GoogleAuthenticatorConfig config = new GoogleAuthenticatorConfig();
 GoogleAuthenticator gAuth = new GoogleAuthenticator();
 GoogleAuthenticatorKey key = gAuth.createCredentials();
 return key.getKey();
 }

 public boolean verifyOtp(String secretKey, String otp) {
 GoogleAuthenticatorConfig config = new GoogleAuthenticatorConfig();
 GoogleAuthenticator gAuth = new GoogleAuthenticator();
 return gAuth.authorize(secretKey, Integer.parseInt(otp));
 }
}

Is code se, hum "GoogleAuthenticator" object create karte hai aur "createCredentials()" method ka use karke OTP generate karte hai. Fir "authorize()" method ka use karke OTP verify karte hai.

Step 3: OTP Generate Karke Response Return Karna
OTP generate karke response return karne ke liye, hum Spring Boot REST API ka use karenge.

java

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class OtpController {
 private final OtpService otpService;

 public OtpController(OtpService otpService) {
 this.otpService = otpService;
 }

 @GetMapping("/generateSecretKey")
 public String generateSecretKey() {
 return otpService.generateSecretKey();
 }

 @GetMapping("/verifyOtp/{secretKey}/{otp}")
 public boolean verifyOtp(@PathVariable String secretKey, @PathVariable String otp)
 {
 return otpService.verifyOtp(secretKey, otp);
 }
}

Is code se, hum "/generateSecretKey" endpoint create karte hai, jisse secret key generate karke response me return karenge. Fir, "/verifyOtp" endpoint create karte hai, jisse secret key aur OTP receive karke OTP verify karenge aur response me boolean value return karenge.

Note: Is code me, maine "GET" method ka use kiya hai, apn apne use case ke hisab se appropriate HTTP method ka use kar sakte hai.

I hope, ye code aapko OTP implementation karne me help karenge.

S

manually generate or verify karna bina dependency ke to ky kare

Agar apn manually OTP generate ya verify karna chahte hai bina kisi third-party library ke, to apn ye steps follow kar sakte hai:

OTP Generate Karne Ke Liye:

1. Generate a random secret key (string) using a cryptographic random number generator.
2. Calculate the current time in Unix time (number of seconds elapsed since January 1, 1970).
3. Divide the current time by the number of seconds that a OTP should be valid (typically 30 seconds).
4. Use an HMAC-SHA1 hash function to generate a 20-byte hash value by passing the secret key and the result from step 3 as inputs.
5. Take the least significant 4 bits of the hash value and use it as an offset to generate a 4-byte integer.
6. Convert the integer to a string of 6 digits using zero-padding if necessary.
7. The resulting string is the OTP.

Sample code for generating OTP manually without any external library:

java

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.time.Instant;

public class OtpService {
 private static final int SECRET_KEY_LENGTH = 20;
 private static final int NUMBER_OF_DIGITS = 6;
 private static final int INTERVAL_SECONDS = 30;

 public String generateSecretKey() {
 SecureRandom secureRandom = new SecureRandom();
 byte[] secretKeyBytes = new byte[SECRET_KEY_LENGTH];
 secureRandom.nextBytes(secretKeyBytes);
 return bytesToHexString(secretKeyBytes);
 }

 public String generateOtp(String secretKey) throws NoSuchAlgorithmException, InvalidKeyException, InvalidParameterException {
 long currentTime = Instant.now().getEpochSecond();
 long timeStep = currentTime / INTERVAL_SECONDS;

 byte[] secretKeyBytes = hexStringToBytes(secretKey);
 byte[] timeStepBytes = ByteBuffer.allocate(Long.BYTES).putLong(timeStep).array();

 SecretKeySpec secretKeySpec = new SecretKeySpec(secretKeyBytes, "HmacSHA1");
 Mac mac = Mac.getInstance("HmacSHA1", secretKeySpec);
 mac.init(secretKeySpec);

 byte[] hash = mac.doFinal(timeStepBytes);

 int offset = hash[hash.length - 1] & 0xf;
 int binaryCode = ((hash[offset] & 0xf) << 24) | ((hash[offset + 1] & 0xf) << 16) | ((hash[offset + 2] & 0xf) << 8) | (hash[offset + 3] & 0xf);

 int otp = binaryCode % (int) Math.pow(10, NUMBER_OF_DIGITS);
 return String.format("%0" + NUMBER_OF_DIGITS + "d", otp);
 }

 private byte[] hexStringToBytes(String hexString) {
 byte[] bytes = new byte[(hexString.length() / 2)];
 for (int i = 0; i < bytes.length; i++) {
 bytes[i] = (byte) Integer.parseInt(hexString.substring(i * 2, i * 2 + 2));
 }
 return bytes;
 }

 private String bytesToHexString(byte[] bytes) {
 StringBuilder sb = new StringBuilder();
 for (byte b : bytes) {
 sb.append(String.format("%02x", b));
 }
 return sb.toString();
 }
}

OTP Verify Karne Ke Liye:

1. Take the user's secret key and the current OTP entered by the user.
2. Calculate the current time in Unix time (number of seconds elapsed since

S

hogy he ya baki he

Mere dwara diye gaye steps aur sample code se apn manually OTP generate aur verify kar sakte hai. Aapko iske liye kisi third-party library ki janur nahi hogi. Agar aapko abhi bhi koi sawaal ya pareshani ho toh apn poochh sakte hai.

Generate response

ChatGPT: Feb 14 Version

Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

https://chat.openai.com/chat

1/1