

싸피 스무고개





왜 이걸 해보나요?

- ✓ Open API를 쓰는 경우가 있을 겁니다
- ✓ 여러분들이 회사에 간다면 Open API를 개발할 수도 있겠죠?
- ✓ 크롤링 같은 것에 활용할 수도 있고요
- ✓ 기본적인 통신에 필요한 객체/라이브러리에 익숙해질 수 있습니다
- ✓ 까만 터미널 화면에 익숙해지는 것도 좋겠죠

API를 호출해봅시다!



✓ 1줄 끝?

```
2000 curl --header "Content-Type: application/json" --request  
POST --data '{"nickname":"kim","yourAnswer":"ssafycial"}' ht  
tp://13.125.222.176/quiz/chopper
```

```
{"code":200,"question":"Q4: 교차 출처 리소스 공유 (Cross Origin Re  
source Sharing)는 추가적인 http 헤더를 이용하여, 한 출처에서 실행  
중인 웹 애플리케이션이 다른 출처의 \n자원에 접근할 수 있는 권한을  
부여하도록 브라우저에 알려주는 것을 말합니다. 자원의 출처가 다르  
다는 것은 3가지 요소를 가지고 파악하는데요. \n3가지 요소에는 domai  
n, port, 그리고 이것이 있습니다. 이것은 무엇일까요? (영문)","next  
Url":"weekend"}ubuntu@ip-172-26-5-70:~$ █
```

객체나 라이브러리를 파악해보자



- ✓ 통신을 위한 많은 종류의 객체나 인터페이스 혹은 라이브러리가 존재합니다
- ✓ 어떤 것을 쓰더라도 통신은 가능할 겁니다
- ✓ 하지만 어떤 종류의 것들이 있고 어떤 장단점이 있는지, 어떤 차이가 있는지 이해를 하고 있는 것이 좋겠죠!
- ✓ 사용하기 전에 내가 원하는 기능을 수행해줄 수 있는 메소드가 있는지 명세를 살펴보도록 합니다

HttpURLConnection



- ✓ A HttpURLConnection with support for HTTP-specific features. See the spec for details.
- ✓ Each HttpURLConnection instance is used to make a single request but the underlying network connection to the HTTP server may be transparently shared by other instances.
- ✓ Since : JDK1.1
- ✓ 굉장히 원시적인 객체이다.

URLConnection

Method Summary

All Methods	Static Methods	Instance Methods	Abstract Methods	Concrete Methods
Modifier and Type		Method and Description		
	abstract void	disconnect() Indicates that other requests to the server are unlikely in the near future.		
	InputStream	getErrorStream() Returns the error stream if the connection failed but the server sent useful data nonetheless.		
	static boolean	getFollowRedirects() Returns a <code>boolean</code> indicating whether or not HTTP redirects (3xx) should be automatically followed.		
	String	getHeaderField(int n) Returns the value for the n^{th} header field.		
	long	getHeaderFieldDate(String name, long Default) Returns the value of the named field parsed as date.		
	String	getHeaderFieldKey(int n) Returns the key for the n^{th} header field.		
	boolean	getInstanceFollowRedirects() Returns the value of this <code>URLConnection</code> 's <code>instanceFollowRedirects</code> field.		
	Permission	getPermission() Returns a <code>SocketPermission</code> object representing the permission necessary to connect to the destination host and port.		
	String	getRequestMethod() Get the request method.		
	int	getResponseCode() Gets the status code from an HTTP response message.		
	String	getResponseMessage()		

RestTemplate



- ✓ Synchronous client to perform HTTP requests, exposing a simple, template method API over underlying HTTP client libraries such as the JDK **HttpURLConnection**, Apache **HttpComponents**, and others.
- ✓ The RestTemplate offers templates for common scenarios by HTTP method, in addition to the **generalized** exchange and execute methods that support of less frequent cases.
- ✓ As of 5.0 this class is in maintenance mode, with only minor requests for changes and bugs to be accepted going forward. Please, consider using the `org.springframework.web.reactive.client.WebClient` which has a more modern API and supports sync, async, and streaming scenarios.

RestTemplate



Method Summary

Methods

Modifier and Type	Method and Description
void	delete (String url, Map<String,?> urlVariables) Delete the resources at the specified URI.
void	delete (String url, Object... urlVariables) Delete the resources at the specified URI.
void	delete (URI url) Delete the resources at the specified URL.
protected <T> T	doExecute (URI url, HttpMethod method, RequestCallback requestCallback, ResponseExtractor<T> responseExtractor) Execute the given method on the provided URI.
<T> ResponseEntity<T>	exchange (String url, HttpMethod method, HttpEntity<?> requestEntity, Class<T> responseType, Map<String,?> urlVariables) Execute the HTTP method to the given URI template, writing the given request entity to the request, and returns the response as ResponseEntity<T>.
<T> ResponseEntity<T>	exchange (String url, HttpMethod method, HttpEntity<?> requestEntity, Class<T> responseType, Object... urlVariables) Execute the HTTP method to the given URI template, writing the given request entity to the request, and returns the response as ResponseEntity<T>.
<T> ResponseEntity<T>	exchange (String url, HttpMethod method, HttpEntity<?> requestEntity, org.springframework.core.ParameterizedTypeReference<T> responseType, Map<String,?> urlVariables) Execute the HTTP method to the given URI template, writing the given request entity to the request, and returns the response as ResponseEntity<T>.
<T> ResponseEntity<T>	exchange (String url, HttpMethod method, HttpEntity<?> requestEntity, org.springframework.core.ParameterizedTypeReference<T> responseType, Object... urlVariables) Execute the HTTP method to the given URI template, writing the given request entity to the request, and returns the response as ResponseEntity<T>.
<T> ResponseEntity<T>	exchange (URI url, HttpMethod method, HttpEntity<?> requestEntity, Class<T> responseType) Execute the HTTP method to the given URI template, writing the given request entity to the request, and returns the response as ResponseEntity<T>.
<T> ResponseEntity<T>	exchange (URI url, HttpMethod method, HttpEntity<?> requestEntity, org.springframework.core.ParameterizedTypeReference<T> responseType) Execute the HTTP method to the given URI template, writing the given request entity to the request, and returns the response as ResponseEntity<T>.
<T> T	execute (String url, HttpMethod method, RequestCallback requestCallback, ResponseExtractor<T> responseExtractor) Execute the HTTP method to the given URI template, preparing the request with the RequestCallback, and reading the response with the ResponseExtractor.
<T> T	execute (String url, HttpMethod method, RequestCallback requestCallback, ResponseExtractor<T> responseExtractor, Object... urlVariables) Execute the HTTP method to the given URI template, preparing the request with the RequestCallback, and reading the response with the ResponseExtractor.
<T> T	execute (URI url, HttpMethod method, RequestCallback requestCallback, ResponseExtractor<T> responseExtractor) Execute the HTTP method to the given URI template, preparing the request with the RequestCallback, and reading the response with the ResponseExtractor.

WebClient



- ✓ **Non-blocking, reactive** client to perform HTTP requests, exposing a fluent, reactive API over underlying HTTP client libraries such as Reactor Netty.
- ✓ Spring을 기준으로 HTTP 요청을 보내기 위해 만들어진 라이브러리들 중에서 가장 최신화된 인터페이스이다.
- ✓ Spring 5.0을 기준으로 유지보수가 이루어지고 있고, 높은 수준의 추상화가 잘 되어있다.



Method Summary

All Methods	Static Methods	Instance Methods	Abstract Methods	Default Methods
Modifier and Type		Method and Description		
static	WebClient.Builder	builder()	Obtain a WebClient builder.	
static	WebClient	create()	Create a new WebClient with Reactor Netty by default.	
static	WebClient	create(String baseUrl)	Variant of create() that accepts a default base URL.	
	WebClient.RequestHeadersUriSpec<?>	delete()	Start building an HTTP DELETE request.	
	WebClient.RequestHeadersUriSpec<?>	get()	Start building an HTTP GET request.	
	WebClient.RequestHeadersUriSpec<?>	head()	Start building an HTTP HEAD request.	
	WebClient.RequestBodyUriSpec	method(HttpMethod method)	Start building a request for the given HttpMethod.	
	WebClient.Builder	mutate()		

추가학습을 해봅시다



- ✓ Javadoc이나 기술 문서(python)를 읽고 메소드와 속성들을 확인해봅니다.
- ✓ 각각의 라이브러리들에는 어떤 차이가 있는지, 어떻게 발전해왔는지를 봅니다.
- ✓ REST의 정의와 개념에 대해서 공부해봅니다.