

Webhook with Spring Boot and H2 Database



by Stoyan Ivanov

(<https://dreamix.eu/blog/author/stoyan-ivanov/>)

August 29, 2017

Dreamix

(<https://dreamix.eu/blog/category/dreamix/>),

frontpage

(<https://dreamix.eu/blog/category/frontpage/>),

Implementations

(<https://dreamix.eu/blog/category/java/implementations/>),

Java

(<https://dreamix.eu/blog/category/java/>)

3 Comments

Want the know-how of experienced developers?

Subscribe for Dreamix Blog now!

Your email address

I WANT THE LATEST IT NEWS!

Recent Posts

- TTJ: JSON Serialization, Functional Interfaces in Java and more | 05/12/22 (<https://dreamix.eu/blog/newsroom/ttj-json-serialization-functional-interfaces-in-java-and-more-05-12-22>)
- Key performance changes in .NET Core 7: an overview (<https://dreamix.eu/blog/tech/net/key-performance-changes-in-net-core-7-an-overview>)
- TTJ: Spring REST Services, JavaFX, Security and more (<https://dreamix.eu/blog/newsroom/ttj-spring-rest-services-javafx-security-and-more>)
- Triple Threat Java 30/11/22 (<https://dreamix.eu/blog/newsroom/triple-threat-java-30-11-22>)
- Triple Threat Java 28/11/22 (<https://dreamix.eu/blog/newsroom/triple-threat-java-28-11-22>)

Categories

(<https://dreamix.eu/blog/category/bizdev>)

>

(<https://dreamix.eu/blog/category/managed-services>)



([/#facebook](#)) ([/#twitter](#)) ([/#linkedin](#))
<https://www.addtoany.com/share?url=https%3A%2F%2Fdreamix.eu%2Fblog%2Fwebhook-with-spring-boot-and-h2-database&title=Webhook%20with%20Spring%20Boot%20and%20H2%20Database>

What exactly is a webhook?

A webhook is one of the best ways for real-time communication with your application. Unlike typical APIs, a webhook delivers data to the applications which means that you get data immediately when it occurs. This makes it much more efficient for both provider and consumer. In the webhook you can register your URLs which it will target when the information is being streamed. A webhook makes a simple HTTP request to your app and you receive data.

Creating your own webhook

For an implementation I use a Spring Boot framework and I store my webhook in H2 database. H2 is an in-memory DB which is recreated every time we start the application. For H2 DB you can find more information here (<http://www.h2database.com/html/main.html>).

In my code I use two fields which I can identify the company with. The first one is a company name. Companies which use a webhook are separated by this field. And the second field is for the type of information which the app will consume. I have added the second field because a company can use particular services which are offered by a webhook.

There is a dto object which represents table from DB. The dto class contains four fields:

1. ID
2. URL
3. company name

4. type.

The URL is the callback which the logic calls when it has finished.
Company name and type:

DTO Code:

```

2
3 import javax.persistence.Column;
4 import javax.persistence.Entity;
5 import javax.persistence.GeneratedValue;
6 import javax.persistence.GenerationType;
7 import javax.persistence.Id;
8 import javax.persistence.Table;
9
10
11
12 @Entity
13 @Table(name = "webhook", schema="webhook")
14 public class WebHook {
15
16     @Id
17     @GeneratedValue(strategy = GenerationType.IDENTITY)
18     private Long id;
19
20     @Column(name="url")
21     private String url;
22
23     @Column(name="company_name")
24     private String companyName;
25
26     @Column(name="type")
27     private String type;
28
29     public WebHook() {
30         super();
31         // TODO Auto-generated constructor stub
32     }
33
34     public WebHook(Long id, String url, String companyName, String type) {
35         super();
36         this.id = id;
37         this.url = url;
38         this.companyName = companyName;
39         this.type = type;
40     }
41
42     //getters and setters
43 }

```

(https://dreamix.eu/blog/category/nearshoring)

(https://dreamix.eu/blog/category/newsroom)

>

(https://dreamix.eu/blog/category/talent)

(https://dreamix.eu/blog/category/startups)

(https://dreamix.eu/blog/category/webothers)

My repository is simple and extends a Crud Repository. I have also added two bonus methods for filtering information easily:

1. The first one finds a webhook both by name and by type of service it has been registered with;
2. The second one is only for company name.

Repository Code:

```

2
3 import java.util.List;
4
5 import org.springframework.data.repository.CrudRepository;
6
7 import eu.dreamix.webhook.dto.WebHook;
8
9 public interface WebhookRepository extends CrudRepository<WebHook, Long>{
10
11     public List<WebHook> findByCompanyNameAndType(String companyName, String type);
12
13     public List<WebHook> findByCompanyName(String companyName);
14 }

```

In my controller I have created one simple method for adding a webhook. It's a simple POST which consumes a webhook in JSON format. I have also attached services for deletion of registration (both by company and by type) and for updating a webhook.

Controller code:

```

2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import org.slf4j.Logger;
7 import org.slf4j.LoggerFactory;
8 import org.springframework.beans.factory.annotation.Autowired;
9 import org.springframework.http.HttpStatus;
10 import org.springframework.http.MediaType;
11 import org.springframework.http.ResponseEntity;
12 import org.springframework.web.bind.annotation.PathVariable;
13 import org.springframework.web.bind.annotation.RequestBody;
14 import org.springframework.web.bind.annotation.RequestMapping;
15 import org.springframework.web.bind.annotation.RequestMethod;
16 import org.springframework.web.bind.annotation.RestController;
17
18 import eu.dreamix.webhook.dto.WebHook;
19 import eu.dreamix.webhook.repository.WebhookRepository;
20
21
22 @RestController
23 @RequestMapping("/webhooks")
24 public class WebHookController {
25     private final static Logger logger = LoggerFactory.getLogger(WebHookController.class);
26
27     private WebhookRepository webHookRepository;
28
29     @Autowired
30     public WebHookController(WebhookRepository webHookRepository) {
31         super();
32         this.webHookRepository = webHookRepository;
33     }
34
35     @RequestMapping(method = RequestMethod.POST,
36         consumes = MediaType.APPLICATION_JSON_VALUE,
37         produces=MediaType.TEXT_MARKDOWN_VALUE)
38     public ResponseEntity<String> addWebHook(@RequestBody WebHook webHook){
39         logger.info("New webhook for " + webHook.getCompanyName() + " is registered");
40         List<WebHook> webhooks = webHookRepository.findByCompanyNameAndType(
41             webHook.getCompanyName(),
42             webHook.getType());
43         if(webhooks != null && webhooks.contains(webHook)){
44             return new ResponseEntity<>("Webhook already exists", HttpStatus.OK);
45         }
46         webHookRepository.save(webHook);
47         return new ResponseEntity<>("Successfully", HttpStatus.CREATED);
48     }
49
50     @RequestMapping(method = RequestMethod.GET, produces=MediaType.APPLICATION_JSON_VALUE)
51     public ResponseEntity<List<WebHook>> getAllWebHooks(){
52         List<WebHook> webhooks = new ArrayList<>();
53         webHookRepository.findAll().iterator().forEachRemaining(webhooks::add);
54         return new ResponseEntity<List<WebHook>>(webhooks, HttpStatus.OK);
55     }
56
57     @RequestMapping(method = RequestMethod.GET,
58         value = "/companies/{companyName}/types/{type}",
59         produces=MediaType.APPLICATION_JSON_VALUE)
60     public ResponseEntity<WebHook> getWebHooksByCompanyNameAndType(
61         @PathVariable String companyName,
62         @PathVariable String type){
63         List<WebHook> webhooks = webHookRepository.findByCompanyNameAndType(
64             companyName, type);
65         return new ResponseEntity<WebHook>(webhooks.get(0), HttpStatus.OK);
66     }
67
68     @RequestMapping(method = RequestMethod.DELETE,
69         value = "/companies/{companyName}/types/{type}",
70         produces=MediaType.TEXT_MARKDOWN_VALUE)
71     public ResponseEntity<String> removeWebHook(
72         @PathVariable String companyName,
73         @PathVariable String type){
74         List<WebHook> webhooks = webHookRepository.findByCompanyNameAndType(
75             companyName, type);
76         if(!webhooks.isEmpty()){
77             webHookRepository.delete(webhooks.get(0));
78             return new ResponseEntity<>("WebHook was successfully deleted.", HttpStatus.OK);
79         }
80         return new ResponseEntity<>("Webhook doesn't exist.", HttpStatus.OK);
81     }
82
83     @RequestMapping(method = RequestMethod.DELETE, value = "/ids/{id}",
84         produces=MediaType.TEXT_MARKDOWN_VALUE)
85     public ResponseEntity<String> removeWebHookById(@PathVariable Long id){
86         WebHook webhook = webHookRepository.findOne(id);
87         if(webhook != null){
88             webHookRepository.delete(webhook);
89             return new ResponseEntity<>("WebHook was successfully deleted.", HttpStatus.OK);
90         }
91         return new ResponseEntity<>("Webhook doesn't exist.", HttpStatus.OK);
92     }
93 }

```

And now that you have your own webhook let's talk about where it is appropriate to use it. Here are my suggestions:

1. When you want to extend your core product with others custom processes which communicate to your business logic in real-time.

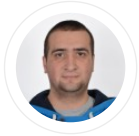


2. When you want to improve user experience and you have an application with asynchronous architecture.

And finally some things you should consider when building a webhook:

1. **Security!** That is one of the most important things that you should consider. You can require the URL to use https.
2. Carefully consider the data that you share .
3. Think about a fallback mechanism in case of a problem with internet traffic. If you don't have a fallback mechanism a webhook can't be used in critical business operations.

Do you have experience on the topic? Feel free to share it in the comments.



Stoyan Ivanov (<https://dreamix.eu/blog/author/stoyan-ivanov>)

Java & Oracle Developer at Dreamix

More Posts (<https://dreamix.eu/blog/author/stoyan-ivanov>) - Website (<http://44>)

([/#facebook](#)) ([/#twitter](#)) ([/#linkedin](#))

(<https://www.addtoany.com/share?url=https%3A%2F%2Fdreamix.eu%2Fblog%2Fwebhook-with-spring-boot-and-h2-database&title=Webhook%20with%20Spring%20Boot%20and%20H2%20Database>)

DO YOU WANT MORE GREAT BLOGS LIKE THIS?

Subscribe for Dreamix Blog now!

I WANT THE LATEST IT NEWS!

6 Comments 1


Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?


Name

Sort by Best ♥ 🔗

 **Abhay Kumar** • 3 years ago

Ridiculuus.. This is a orm example how to persist, its not a example which demonstrates webhook concept.


14 ^ | ▼ • Reply • Share ›

 **ramasani subramanyam** ➔ Abhay Kumar • 10 months ago

Hi bro have you implemented webhook please help me how to.


ramasani17@gmail.com ping me bro very urgent for me

^ | ▼ • Reply • Share ›

 **Marlon Antonio Medina Taguican** • 3 years ago


you just created an API and named it Webhook

9 ^ | ▼ • Reply • Share ›

 **PAMELA ANOU** • 5 years ago • edited


I'm looking for how created a webhook!! but you're not creating a webhook you managed and use it!!! the title "Creating your own webhook" its not for this tuto! :((((

9 ^ | ▼ 1 • Reply • Share ›

 **Yash bhardwaj** • 5 months ago

This is nothing related to webhooks, Dreamix must remove this blog, as it's not explaining what it claims and is a misleading one.

1 ^ | ▼ • Reply • Share ›

 **Stephen A. Olujare** ➔ Yash bhardwaj • 5 months ago

Home (<https://dreamix.eu/blog/>) » Webhook with Spring Boot and H2 Database

(index.html)

Bespoke software development



(<https://www.linkedin.com/company/dreamix-group/>) (<https://www.facebook.com/dreamixgroup/>) (<https://twitter.com/dreamixgroup>) (<https://plus.google.com/u/02938918336185185480>)

Blog

- Timeout Function in Python 3 (<https://dreamix.eu/blog/webothers/timeout-function-in-python-3>)
- Webhook with Spring Boot and H2 Database (<https://dreamix.eu/blog/java/webhook-with-spring-boot-and-h2-database>)
- Implementing custom authorization function for Spring's @Pre and @Post annotations (<https://dreamix.eu/blog/java/implementing-custom-authorization-function-for-springs-pre-and-post-annotations>)

- [Reactive Java benchmark. Vert.x vs. Spring Framework 5 \(https://dreamix.eu/blog/java/reactive-java-vert-x-vs-spring-framework-5\)](https://dreamix.eu/blog/java/reactive-java-vert-x-vs-spring-framework-5)
- [What needs to be unit tested. How much code coverage is needed. \(https://dreamix.eu/blog/java/what-needs-to-be-unit-tested-how-much-code-coverage-is-needed\)](https://dreamix.eu/blog/java/what-needs-to-be-unit-tested-how-much-code-coverage-is-needed)
- [How to use Jython with Spring Boot \(https://dreamix.eu/blog/java/how-to-use-jython-with-spring-boot-2\)](https://dreamix.eu/blog/java/how-to-use-jython-with-spring-boot-2)
- [How to setup a project with Angular 7 and Spring Boot, that provides custom embeddable Angular components via Angular Elements \(https://dreamix.eu/blog/frontpage/how-to-setup-a-project-with-angular-7-and-spring-boot-that-provides-custom-embeddable-angular-components-via-angular-elements\)](https://dreamix.eu/blog/frontpage/how-to-setup-a-project-with-angular-7-and-spring-boot-that-provides-custom-embeddable-angular-components-via-angular-elements)
- [All blog posts \(https://dreamix.eu/blog/\)](https://dreamix.eu/blog/)

Categories

- [All \(https://dreamix.eu/blog/\)](https://dreamix.eu/blog/)
- [Business Development \(https://dreamix.eu/blog/category/bizdev\)](https://dreamix.eu/blog/category/bizdev)
- [Managed Services \(https://dreamix.eu/blog/category/managed-services\)](https://dreamix.eu/blog/category/managed-services)
- [Nearshoring \(https://dreamix.eu/blog/category/nearshoring\)](https://dreamix.eu/blog/category/nearshoring)
- [Newsroom \(https://dreamix.eu/blog/category/newsroom\)](https://dreamix.eu/blog/category/newsroom)
- [People \(https://dreamix.eu/blog/category/talent\)](https://dreamix.eu/blog/category/talent)
- [Startups \(https://dreamix.eu/blog/category/startups\)](https://dreamix.eu/blog/category/startups)
- [Web and others \(https://dreamix.eu/blog/category/webothers\)](https://dreamix.eu/blog/category/webothers)
- [Company Culture \(https://dreamix.eu/blog/tag/company-culture\)](https://dreamix.eu/blog/tag/company-culture)
- [Java Daily \(https://dreamix.eu/blog/tag/java-daily\)](https://dreamix.eu/blog/tag/java-daily)

Java

- [All \(https://dreamix.eu/blog/category/java\)](https://dreamix.eu/blog/category/java)
- [Implementations \(https://dreamix.eu/blog/category/java/implementations\)](https://dreamix.eu/blog/category/java/implementations)
- [JBoss \(https://dreamix.eu/blog/category/java/jboss\)](https://dreamix.eu/blog/category/java/jboss)
- [News \(https://dreamix.eu/blog/category/java/javanews\)](https://dreamix.eu/blog/category/java/javanews)
- [Server \(https://dreamix.eu/blog/category/java/updates\)](https://dreamix.eu/blog/category/java/updates)
- [XML \(https://dreamix.eu/blog/category/java/xml-java\)](https://dreamix.eu/blog/category/java/xml-java)
- [Java implementations \(https://dreamix.eu/blog/category/java-implementations\)](https://dreamix.eu/blog/category/java-implementations)

Oracle Fusion Middleware

- [All \(https://dreamix.eu/blog/category/oracle-2\)](https://dreamix.eu/blog/category/oracle-2)
- [Oracle ADF \(https://dreamix.eu/blog/category/oracle-2/oracle-adf\)](https://dreamix.eu/blog/category/oracle-2/oracle-adf)
- [Oracle BPM \(https://dreamix.eu/blog/category/oracle-2/bpm\)](https://dreamix.eu/blog/category/oracle-2/bpm)
- [Oracle Database \(https://dreamix.eu/blog/category/oracle-2/oracle-database\)](https://dreamix.eu/blog/category/oracle-2/oracle-database)
- [Oracle Forms Migration \(https://dreamix.eu/blog/category/oracle-2/oracle-forms-migration\)](https://dreamix.eu/blog/category/oracle-2/oracle-forms-migration)
- [Oracle SOA \(https://dreamix.eu/blog/category/oracle-2/soa\)](https://dreamix.eu/blog/category/oracle-2/soa)
- [Oracle WebCenter \(https://dreamix.eu/blog/category/oracle-2/oracle-webcenter\)](https://dreamix.eu/blog/category/oracle-2/oracle-webcenter)

Services

- [Custom Solutions \(/services/custom-software-solutions\)](/services/custom-software-solutions)
- [Nearshore Development \(/services/nearshore-software-development-outsourcing\)](/services/nearshore-software-development-outsourcing)
- [Dedicated Team \(/services/hire-dedicated-software-development-team\)](/services/hire-dedicated-software-development-team)





- [Managed Services \(/services/managed-services\)](/services/managed-services)
- [BPM \(/services/business-process-management-software-solutions\)](/services/business-process-management-software-solutions)
- [Testing and QA \(/services/software-testing-and-qa\)](/services/software-testing-and-qa)
- [Software Architecture \(/services/software-architecture-development\)](/services/software-architecture-development)

Technologies

- [Java EE \(/technologies/partner-with-java-ee-development-company\)](/technologies/partner-with-java-ee-development-company)
- [AngularJS \(/technologies/angularjs-application-development-outsourcing\)](/technologies/angularjs-application-development-outsourcing)
- [Oracle ADF \(/technologies/oracle-adf-development\)](/technologies/oracle-adf-development)
- [Oracle SOA and BPM \(/technologies/oracle-soa-and-bpm-development\)](/technologies/oracle-soa-and-bpm-development)

Contact us

-  [Email us \(/contact-us\)](/contact-us)
-  [\(+359\) 884-116-309 \(tel:+359884116309\)](tel:+359884116309)