



Creative Commons Attribution –  
NonCommercial 4.0 International License

Professional paper

<https://doi.org/10.31784/zvr.7.1.10>

Received: 15 January 2019

Accepted: 7 March 2019

# A COMPARISON OF BACK-END FRAMEWORKS FOR WEB APPLICATION DEVELOPMENT

**Marin Kaluža**

PhD, Senior Lecturer, Polytechnic of Rijeka, Vukovarska 58, 51 000 Rijeka, Croatia; e-mail: mkaluza@veleri.hr

**Marijana Kalanj**

Student, Polytechnic of Rijeka, Vukovarska 58, 51 000 Rijeka, Croatia; e-mail: mkalanj@veleri.hr

**Bernard Vukelić**

PhD, Senior Lecturer, Polytechnic of Rijeka, Vukovarska 58, 51 000 Rijeka, Croatia; e-mail: bvukelic@veleri.hr

## ABSTRACT

*Web applications have a complex structure, and for more efficient and faster writing of original program code, frameworks are often used. There are numerous different frameworks on the market that are used to build different parts of software architecture. Only back-end frameworks for web applications development are analyzed. Previous research on evaluation criteria for back-end frameworks, and various sources which by various reasons, represent the list of popularity of back-end frameworks are shown. The framework selection procedure is described, and the reasons of framework selection (Laravel, Rails, Django and Spring) for the analysis are given. The examination (verification) and evaluation of the degree of satisfaction of the selected basic and additional comparison criteria in selected frameworks are carried out. The analysis shows insignificant differences in the overall ratings of analyzed frameworks according to the basic comparison criteria. According to the additional comparison criteria, the analysis shows significant differences in the overall ratings of analyzed frameworks. It is shown that the Django framework, because it is suitable for development of larger applications, suitable for beginner developers, and achieves the highest overall rating of additional comparison criteria, is the best choice of analyzed back-end frameworks for web application development.*

**Key words:** framework, back-end, comparison criteria

## 1. INTRODUCTION

A web application consists of a client part that is executed on a user's web browser, and a server part that is executed on a web server. The web server executes various processes that create the document in HTML format and deliver to the client who then interprets and displays HTML

content. When designing a web application, the question arises as to which technology to choose. Using of framework (FW) in the development process could increase a level of quality and standardization of web applications. Today, there are a variety of FWs on the market that help to create web applications in various ways. The paper analyzes and compares back-end FWs.

By analyzing and evaluating various criteria which are essential for building and maintaining web applications, the advantages and disadvantages of frequently used back-end FWs are shown. The objective is to use analysis and evaluation in order to show which FWs are more suitable for web application development.

The paper describes the criteria relevant to the comparison of FWs, the FWs are analyzed according to the described criteria, and the FW analysis results are compared according to the described criteria.

When researching basic concepts related to the back-end FW, it can be observed that there is a large number of FWs on the market and the question is why it is so. Each FW is different and contains some advantages and shortcomings in relation to the other. There is a number of criteria that are important when selecting a FW, so this paper analyses criteria for the selected FWs. Some FWs are more suitable for smaller projects, while others are customized for larger and more demanding projects. Some FWs are safer than the others, while others have documentation of better quality. FW ratings by criteria can be used in the selection process of the appropriate FW for web application development.

## **2. RELATED WORK**

There are multiple sources about the criteria that are essential for FW selection. One back-end comparison matrix FW is Raible Matrix (Diaz, 2013) adapted for back-end FWs. The table explains 20 criteria that are relevant to the overall picture of a particular FW (Raible, 2010a). It explains each comparison criterion, evaluates the criteria for each FW, and the method for evaluating the criteria.

Time for learning, technical characteristics, community engagement and the consequence of choice are the most important features when choosing a FW (Varisco, 2016). It is considered that time for learning is a very important factor regardless of which FW is used, which means that each FW requires a certain amount of time for learning. The basic technical features that should be checked before choosing a FW are: performance, compatibility and whether the FW is appropriate for the type of application structure used. Community engagement is important, i.e. the data about the number of developers who use a certain FW and the number of discussions related to this matter (Varisco, 2016).

In Symfony (<https://symfony.com>), similar criteria which are used for FW selection, such as community popularity and size, support, licenses, documentation, and resource availability on the market are listed.

An important factor for technology selection is TTM (Time to Market) (Maheedharan, 2017). One of TTM's features suggests choosing technologies which have the ability to integrate with third-party solutions to simplify the integration process (Maheedharan, 2017).

According to GitHub starback backgammon, the back-end FW with the highest score of 47.5k is Laravel, followed by Ruby On Rails with 41.5k stars, Django with 37k stars, and Spring with 31.5k stars. Back-end FWs mentioned in the GitHub repository after the above mentioned, and having over 10k stars are Symfony and Yii2 (<https://github.com>)

Hotframeworks (<https://hotframeworks.com>) shows the FW rating according to GitHub and Stackoverflow results, together with the final result. The Back-end FW with the highest score is Ruby On Rails, followed by Django, Laravel and Spring. Back-end frameworks which are at the top of the list are Symfony, Flask, CodeIgniter, Zend, Yii (<https://hotframeworks.com>).

An FW has become an indispensable part of web application development, and user requirements are always on the rise. Consequently, more complex technologies which are used for web application development are required. The most powerful and most interesting FWs in 2018 are: Express, Django, Ruby On Rails, Laravel and Spring (Goel, 2018).

FWs consist of tools and languages used in server-side development. For the selection of the right technology, in addition to knowledge of databases and programming languages, understanding of server architecture is essential. Rails, Laravel, Django, Phoneix, Flask, Pyramid and Express were the most popular frameworks in 2018 (Tripathi, 2018).

Stackshare (<https://stackshare.io>) website provides information on software and website creation tools which are currently the best. It investigates and compares tools, as well as provides insights into the tools used by the most well-known companies. According to their comparison, 6620 participants stated that they used Rails FW, Django 4660, Spring 1520, and Laravel 4000 (<https://stackshare.io>).

It is stated that according to Hacker News, Rails has 2370 points, Django 2490, Spring 332 and Laravel 516 points. They, also, say that according to Reddit, Rails has 1960 points, Django 5090, Spring 2730 and Laravel 3160 points (<https://stackshare.io>).

According to the research (<https://stackshare.io>), developers usually choose Rails because of their rapid development, excellent so-called "gems", large community (documentation, tutorials...) and MVC access. Django is also used due to rapid development, large community, open source, MVC access, and ease of learning. Spring FW is most commonly used due to Java, because it is open source, has a large community, it is said to be extremely powerful, and because of the companies. Laravel is most often used for pure architecture, growing community, because it is composer friendly and because it is very powerful for PHP.

The Medium Corporation website includes a comparison of Laravel and Spring FWs by a developer who first used Laravel, then Spring. According to the statements, Laravel is easy to use, has a clear syntax, powerful documentation and security concepts, and has all the predispositions for beginner developers (Audurus, 2018). Spring FW has a more extensive documentation, especially

for security, and takes more time to learn and read documentation than Laravel (Audurus, 2018). Spring FW is better for larger applications and enterprise applications (Audurus, 2018).

The main advantages of Spring FW are a large community, high scalability, quality of documentation, and a broad ecosystem (<https://slant.io>, 2018). The main advantages of Laravel are: built-in ORM, its own CLI, good for building RESTful API, and good quality documentation (<https://slant.io>). The advantages of Rails are: small project development is very simple, and it is possible to finish it in a very short time, many plugins (“gems”) are available, large community, nice and readable codes (<https://slant.io>) are available. The advantages of Django are: quick and easy prototype development, it is a mature FW with many plugins that have been developed over the years, very customizable, admin panel and ORM are “out of the box” (<https://slant.io>).

A shortcoming of Django is that it can be non-adjustable to small applications (<https://slant.io>). The shortcomings of Rails are: too many dynamic changes in the background, not a popular language outside web development, learning curves appear simple in the beginning, but turn out to be complicated, too many conventions, poor performance (<https://slant.io>). Laravel’s disadvantages are the following: too many unnecessary folders and files, too many “magic methods”, difficult to use model properties, confusing learning curves - many features are updated over time, obsolete tutorials, bad design practices and poor performance (<https://slant.io>). Spring’s shortcomings are: the lack of UI development, possibly complicated for beginners, complex and slow prototyping (<https://slant.io>).

The Google Trend criterion originated according to Google Trends web site data in the last 5 years. The most searched FW is Spring. Spring and Laravel have a growing Google Trend (<https://trends.google.com>).

### **3. RESEARCH METHODOLOGY**

FWs comparison criteria were selected according to the above mentioned sources. Most of the basic criteria have been selected according to the Raible Matrix (Raible, 2010a), and they are: developer’s productivity, quality of documentation / tutorials / stackoverflow issues (project health), developer perception, developer availability, job trends, templates, plugins/ add-ons, I18n and I10n , testing, validation, multi-language support, cell phone / iPhone, and degree of risk. According to Symphony (<https://symfony.com>), the FW license criteria were selected, and the ability to integrate with third-party solutions.

The criteria will be described using the descriptive method. Based on the description of the criteria, a comparison and rating of each FW will be made. Ratings for each criterion will range from 0 to 1. Rating 0 is used when a FW does not meet the criterion, rating 0.5 when it partially meets the criterion, and rating 1 when FW fully meets the criterion.

Activities of research methodology are:

1. Literature review
2. Selection of basic criteria for comparison FWs

3. Choosing FWs for comparison
4. Literature review (whether FWs support basic criteria)
5. Evaluation of the FW (based on basic criteria)
6. Selection of additional criteria
7. Literature review (whether FWs support additional criteria)
8. Evaluation of the FW (based on additional criteria)

Basic criteria were selected based on previous research. Most comparison criteria were selected according to Raible's matrix.

Additional criteria were selected after the basic. Additional criteria are evaluations of the users of selected FWs. These are ratings from GitHub, Reddit, Stack Overflow, Hacker News, and Google Trend. On the StackShare, Inc. and Slant.io pages, you can select the FWs for comparison to find advantages, disadvantages between FWs, and other features of the FWs.

FWs selection was based on FWs popularity from GitHub (<https://github.com>), HotFrameworks (<https://hotframeworks.com>), and Tripathi (2018). According to them, the most used FWs for web application development are: Ruby on Rails, Django, Spring and Laravel.

### 3.1 BASIC CRITERIA FOR FW COMPARISON

Most of the criteria were selected from (Raible, 2010a), Symphony (<https://symfony.com>), and other experts mentioned in related work.

#### 1. Code generator

Code generating can be used for small parts of code or for entire applications; it can be used in different programming languages and using different techniques (Tomassetti, 2018).

The intelligent code generator for Laravel allows you to generate views, controllers, routes, migrations, languages, and form requests. It is flexible and customizable, and it covers many uses. Code generator installation is simple, and there are detailed instructions on installing and using the code generator on the official website (<https://github.com>).

FW Ruby On Rails owns a code generator. Using the command "rails generate" it is possible to obtain a list of all generators owned by FW, together with their detailed description. The most important Rails generators are a generator for controller, model, migration, and it is also possible to create one's own generator (<https://guides.rubyonrails.org>).

Django FW has multiple code generators. Genius is a flexible code generator for Django FW. It is compact and language structure is easy to write and read, it is independent and has a good documentation. Genius generates a model, display, REST Api, and application templates which are customizable (<https://genius-project.io>).

Germ is the code generator for Spring FW. It generates the standard MEAVEN project structure, generates configuration files, model, repository, services, and controllers (<https://docs.spring.io>).

All FWs mentioned above have the ability to automatically generate code and all FWs are rated 1.

## 2. *Project health*

According to Raible (2010a), the project health criterion implies the number of books and the number of tags on the StackOverflow page. According to Varisco (2016), it implies the number of people on the Internet who are talking about it, and who use technology – the official documentation. The more recognizable and well-known FW will develop more and live longer (<https://symfony.com>).

Laravel FW's official pages feature an extensive and complete documentation. There is also a community on the Laracast website with various tutorials, questions and discussions related to FW. There are numerous tutorials available on Youtube, and the Stackoverflow site has about 104,327 tags for Laravel (<https://laracasts.com>).

On the official Ruby on Rails FW website, there is an official documentation which guides through the process of installing and working with the FW. There are many tutorials available on Youtube, and Stackoverflow has around 300,699 tags (<https://guides.rubyonrails.org>).

Django FW official site contains documentation with detailed FW instructions, Stackoverflow has about 186 288 tags, and there are numerous tutorials available on Youtube (<https://docs.djangoproject.com>).

Spring FW official site contains training documentation, projects and guidelines for building a high-quality application. There are about 144,334 tags on the Stackoverflow page (<https://spring.io>).

According to the data obtained by FW research, all the listed FWs are rated 1 because they have numerous sources, a high-quality documentation and a large number of tags on the Stackoverflow page.

## 3. *Developer's perception*

According to Raible (2010a), the developers' perception criterion implies developer's satisfaction. The level of developer's satisfaction in relation to a particular FW is evaluated.

According to GitHub (<https://github.com>), HotFrameworks(<https://hotframeworks.com>, 2018), and Goel (2018), this paper analyzes the best back-end FWs on the market developers are most satisfied with, thus each analyzed FW was rated 1.

## 4. *Developer's availability*

According to Raible (Raible, 2010a), the rating for developer's availability criterion is based on the number of developers with a FW listed as a skill on their LinkedIn profile.

According to LinkedIn, a total of 192,680 users listed Laravel, 278,856 listed Ruby On Rails, 212,681 listed Django and 530,211 listed Spring FW (<https://www.linkedin.com>).

According to LinkedIn, Spring FW has the largest number of available developers, that is, even twice more than any other FW. Therefore, Spring FW is rated 1. Other FWs are rated 0.5.

#### 5. *Business trends*

According to Raible (2010a), the criterion is based on the number of jobs offered at Dice.com or Indeed.com. The rating for the business trend criterion is based on the number of jobs offered.

According to Dice.com, Laravel offers 152 job positions, Django counts 353 positions, Ruby On Rails 549 positions, and Spring 4,619 positions (<https://www.dice.com>).

Spring FW has significantly more positions than other FWs, therefore it is rated 1, while the other FWs are rated 0.5.

#### 6. *Templates*

According to Raible (2010a), the criterion is based on the number of templates available for a particular FW. The criteria rating for the template criterion is based on whether a particular FW template is available or not.

Laravel owns a very simple and useful Blade template. The primary advantages of using Blade are the inheritance of templates and sections (<https://laravel.com>). Django FW also owns a template called DTL (Django Template Language), as well as a popular alternative Jinja2, but third-party templates can also be used (<https://www.djangoproject.com>). Spring owns many template libraries such as: JSP, Tiles, Thymeleaf, Freemarker and Velocity (<https://docs.spring.io>). Ruby On Rails Templates are simple Ruby files which contain DSL to add initialization to a new or existing Rails project (<https://guides.rubyonrails.org>).

All the analyzed FWs have templates, thus they are rated 1.

#### 7. *I18n and l10n*

According to Raible (2010a), the criterion is based on FW support for i18n and l10n.

Laravel's localization features provide a convenient way to reach strings in different languages, enabling an easy support for multiple languages within the app. The language rows are stored in files within the resources/ language directory. Within that directory, there must be a subdirectory for each language which the application supports (<https://laravel.com>).

Django FW has full support for text translation, date formatting, time and numbers, and time zone. It allows developers and template authors to determine which parts of applications should be translated or formatted for local languages and cultures. It allows localization of web applications to specific users according to their desires (<https://www.djangoproject.com>).

The Ruby I18n gem that comes with Ruby on Rails provides a simple and expandable multi-language translation framework (<https://guides.rubyonrails.org>).

Spring FW also follows the form and provides support for I18n and I10n using Spring Interceptors, Locale Resolvers and Resource Bundles for different locations (Pankaj, 2016).

Since all FWs analyzed support I18n and I10n, they are rated 1.

## 8. *Testing*

According to Raible (2010a), FW rating is based on whether the test support is embedded in FW or not.

Laravel has a built-in testing support, that is testing support comes with the PHPUnit, and the testing file is set up for the application. Laravel FW is also supplied with practical methods that enable application testing (<https://laravel.com>).

Ruby on Rails has a built-in test support. Rails creates the test directory as soon as the Rails project is created (<https://guides.rubyonrails.org>).

For testing, Django FW uses unittest modules which are built into the Python standard library. Any other Python FW can be used for testing, but Django FW itself does not support testing (<https://www.djangoproject.com>).

Spring FW has a built-in test support, a built-in test package, and a comprehensive set of temporary objects which are useful for web context, controller, and filter testing (<https://spring.io>).

Test criteria rating is based on whether test support is embedded in FW or not. All the analyzed FWs have a built-in test support, except Django FW, which has an indirect testing support. Therefore, all the analyzed FWs are rated 1.

## 9. *Validation*

According to Raible (2010a), the criterion is based on (non)existence of validation support.

Laravel offers several different approaches for checking incoming application data. By default, Laravel's basic controller class uses the ValidatesRequests feature which provides a convenient method for verifying the incoming HTTP request with a variety of powerful validation policies (<https://laravel.com>). On its official pages, under the chapter validation Ruby On Rails explains in detail the validation process (<https://guides.rubyonrails.org>). Django models support the use of simple and useful functions and classes known as validators. Validator is an object or function which can be called and which takes the value and returns nothing if the value is valid, and if it is not, it sends a validation error (<https://www.djangoproject.com>). Spring FW includes a validator interface which can be used in each application layer (<https://spring.io>).

Criteria rating is based on FW support for validation. All FWs ensure the validation process and assign them rate 1.



#### 10. *Mobile phone / Iphone support*

According to Raible (2010a), the rating for mobile / iPhone support criterion is based on FW mobile support. All the analyzed FWs have the mobile phone / iPhone support and each is rated 1.

#### 11. *Licensing*

According to Symphony (<https://symfony.com>), licenses are very important because they can significantly affect applications.

Rails and Laravel have MIT (<https://mit-license.org>), and Spring and Django have BSD licenses (<http://www.linfo.org>, 2018). None of the licenses limits the application and the application is not subject to licenses, so the rating of all the analyzed FWs in terms of license criterion is 1.

#### 12. *The possibility of integration with third-party solutions*

According to Maheedharan (2017), the criterion of the ability to integrate with third-party solutions refers to the ability of an FW to integrate one or more parts of the application with the existing third-party solutions.

Laravel FW has the ability to integrate third-party solutions using the API, and converts incoming response into a consistent data structure (Ochoa, 2018). Ruby On Rails FW also uses the API for integration with third-party solutions, and a detailed integration process (Gonçalves, 2017) is explained on the official pages. Django FW uses the third-party integration API (Stein, 2017). Spring FW enables third-party integration via adapter (<https://spring.io>).

All the analyzed FWs, in terms of the criterion of integration with third-party solutions, are rated 1.

#### 13. *Multi-language support*

According to Raible (2010a), the rating for multi-language support criterion is based on the ability to write core elements in multiple languages (e.g. Java, Groovy, Scala,...).

Spring FW is the only FW with multi-language support; it supports Java, Groovy and Kotlin. Other analyzed FWs do not have multi-language support, therefore Spring FW is rated 1, and the others are rated 0 (<https://spring.io>).

#### 14. *Plugins/add-ons*

According to Raible (2010a), the rating for the plugins / add-ons criterion is based on whether a FW is able to set plugins (extensions).

Laravel offers plugin functionality and is installed using a command within a particular project (<https://github.com>). Django plugins offer functionality in order for the applications which are made with FW Django can be reused (<https://docs.djangoproject.com>). Ruby On Rails FW uses the rails generate plugin command to create a plugin within the project (<https://guides.rubyonrails.org>).

Django, Ruby On Rails and Laravel have the option of setting up plugins, so they are rated 1. Spring FW does not have the option to set plugins, so it is rated 0.

### 3.2 ADDITIONAL CRITERIA FOR FW COMPARISON

After the basic criteria analysis, there is no significant difference between the FWs. Additional criteria are mostly oriented to the opinions and ratings of developers, that is StackShare (<https://stackshare.io>), Audurus (2018) and Slant (<http://www.slant.io>).

#### 15. *Customized for large applications*

According to Audurus (2017), Spring is more adapted to the existing applications, while Laravel is not. Also, Django is tailored to larger applications, while Rails is better for smaller applications.

Accordingly, FWs Spring and Django are rated 1, and FWs Laravel and Rails are rated 0.

#### 16. *High Scalability*

According to the grades provided by StackShare (<https://stackshare.io>), Spring is highly scalable, followed by Django, while Rails and Laravel are less scalable.

Accordingly, FWs Spring and Django are awarded a rating of 1, and FWs Laravel and Rails are awarded a rating of 0.

#### 17. *Adapted to beginners*

According to the rates of StackShare (<https://stackshare.io>) Rails, Laravel and Django are more adapted to beginners, and are rated 1, while Spring FW is not recommended for beginners and is rated 0.

#### 18. *Good for business applications development*

FWs Spring and Django are, according to StackShare (<https://stackshare.io>) very good for building enterprise applications and are therefore rated 1, while FWs Rails and Laravel are not recommended for this type of app and are rated 0.

#### 19. *Rapid prototype development*

According to Slant (<http://www.slant.io>), Django and Laravel FWs are good for prototype development, therefore they are rated 1, while FWs Rails and Spring are not good enough to build an application prototype. and are therefore rated 0.

#### 20. A growing Google Trend

Rate 1 for this criterion is given to FW Laravel because, according to Google Trends (<https://trends.google.com>), there is a growing trend. Rate 0 is given to the other analyzed FWs, since, according to (<https://trends.google.com>, 2018), they have a downward trend.

#### 21. Points by Haker News users

Rate 1 for this criterion is given to FW Django because, according to StackShare (<https://stackshare.io>), it has the highest score. Rate 0.5 is given to other analyzed FWs.

#### 22. Points by Reddit users

Rate 1 for this criterion is given to FW Django because, according to StackShare (<https://stackshare.io>), it has the highest score. Rate 0.5 is given to other analyzed FWs.

#### 23. Star Rating on GitHub

Rate 1 for this criterion is given to FW Laravel because, according to GitHub (<https://github.com>), it has the highest score. Rate 0.5 is given to other analyzed FWs.

#### 24. The number of Stack Overflow tags

Rate 1 for this criterion is given to FW Rails because, according to StackShare (<https://stackshare.io>), it has the most tags. Rate 0.5 is given to other analyzed FWs.

## 4. RESULTS AND DISCUSSION

The scores of analyzed FWs for basic criteria are shown in Table 1.

Table 1. FW comparison according to the basic criteria

Criterion	Laravel	Ruby On Rails	Django	Spring
1. Code generator	1	1	1	1
2. Project health	1	1	1	1
3. Developer's perception	1	1	1	1
4. Developer's availability	0.5	0.5	0.5	1
5. Business trends	0.5	0.5	0.5	1
6. Templates	1	1	1	1
7. l18n and l10n	1	1	1	1
8. Testing	1	1	1	1
9. Validation	1	1	1	1
10. Mobile phones / iPhone support	1	1	1	1
11. Licensing	1	1	1	1

12. The possibility of integration with third-party solutions	1	1	1	1
13. Multi-language support	0	0	0	1
14. Plugins/add-ons	1	1	1	0
TOTAL	12	12	12	13

Source: Author

The table above shows apparently that most FWs meet the main criteria needed to build a web application. The differences are in relation to criterion 4. In terms of the availability of developers, Spring FW has significantly more developers who listed Spring as a skill, and in terms of criterion 5, it has the most open positions for the job. Spring FW differs from the rest and in relation to criterion 14 (plugins / add-ons), because there is no plugin option. Spring FW received the highest rate because of the criteria which do not directly affect the choice of a FW, but are the criteria that determine the market situation. The criteria used to evaluate FWs are the relevant criteria for assessing FWs. There are no major differences among the FWs analyzed in meeting the basic criteria.

After the baseline comparison, the existing sources of comparison of the selected FWs were analyzed, which provided the additional results. The scores of analyzed FWs for additional criteria are shown in Table 2.

Table 2. FW comparison according to the additional criteria

Criterion	Laravel	Ruby On Rails	Django	Spring
1. Customized for large applications	0	0	1	1
2. High scalability	0	0	1	1
3. Adapted to beginners	1	1	1	0
4. Good for businesses applications development	0	0	1	1
5. Rapid prototype development	1	0	1	0
6. A growing Google Trend	1	0	0	0
7. Points by Haker News users	0.5	0.5	1	0.5
8. Points by Reddit users	0.5	0.5	1	0.5
9. Star Rating on GitHub	1	0.5	0.5	0.5
10. The number of Stack Overflow tags	0.5	1	0.5	0.5
TOTAL	5.5	3.5	8	5

Source: Author

Table 2 shows the additional criteria for evaluating the analyzed FWs. Django FW scored the highest overall score according to these criteria. In relation to the results obtained, it can be said that the FWs Rails and Laravel are similar. They are tailored to the development of smaller applications,

less scalable than the other two analyzed FWs, and good for beginners. Laravel stands out with a growing Google Trend and the highest score of GitHub stars among the FWs. Rails stands out for having the most Stack Overflow tags. Unlike the previous two FWs, Django and Spring are more tailored to build larger applications and applications for business, and are more scalable. Spring FW stands out for its high scalability. Django has the highest score from the Hacker News and Reddit web site users, it is good for rapid prototype development.

The limitations of the research are: (1) all criteria are equally important, (2) the frameworks are evaluated with only three values (0, 0.5 and 1), (3) influences between the criteria is not included.

Future research might include:

1. weighting the criteria by using some of the methods: direct assessments, AHP (Gligora Marković, et al., 2018), ANP (Kadoić, et al., 2017b)
2. evaluating FWs on the scale [0;1] - all values between 0 and 1
3. using a more sophisticated method for problem analysis which enables modelling influences between criteria, ex. ANP (Saaty, 2001), SNAP (Kadoić, et al., 2017a).

Future research should include more experts in the FW evaluation process.

## **5. CONCLUSION**

According to previous research, some criteria are critical to the framework evaluation. Evaluation and comparison of Django, Rails, Spring and Laravel framework was performed using the selected criteria. All the selected frameworks generally meet all the basic criteria, and have shown little differences. Developer availability and business trends criteria are better rated for Spring compared to other frameworks. Spring is the only FW which supports the ability to write the original program code in multiple programming languages. Also, Spring does not support the ability to implement plugins, which may mean a slower development because in case of implementing some of the functionality that is generally available, it will be more difficult to implement it in the application. The most noteworthy overall rating of the basic comparison criteria is Spring.

Due to the small differences between the analyzed frameworks, additional comparison criteria were defined. For larger applications and applications for business systems, Spring and Django frameworks are more suitable, since they are more scalable than the other analyzed frameworks. Spring has a high-quality and extensive documentation that takes more time to study and is not suitable for beginners, but is the oldest in this group of frameworks, so it is reliable. Django is more adapted to beginners than Spring, and it is good for building a prototype application, but has somewhat poorer documentation than Spring. Unlike Spring and Django, Rails and Laravel are suitable for smaller applications and have simpler documentation. Django has the best additional rating of additional criteria.

According to the basic comparison criteria, Spring has insignificantly the best rates, and according to additional criteria, Django has the highest scores. According to the additional comparison

criteria, the choice of a framework depends on the requirements of the application being built, and as Django is suitable for the development of larger applications, and is suitable for beginner developers, Django is the best choice among the analyzed back-end frameworks for web application development.

## REFERENCES

- Audurus, (2018) Spring vs Laravel, <https://medium.com/@audruis/spring-vs-laravel-1af76c35d855> (15. 12. 2018)
- Diaz, C., (2013) "A web framework comparasion: Matt Raible's opinion", <http://blog.websitesframeworks.com/2013/03/web-framework-comparison-matt-raible-opinion-138/> (21.11.2018.) <https://doi.org/10.1016/j.cobeha.2018.04.011>
- Goel, A., (2018) "Top 10 Web Development Frameworks in 2018", <https://hackr.io/blog/top-10-web-development-frameworks-in-2018> (28. 11. 2018.)
- Gonçalves, A., (2017) "Integrating a Third Party API with Rails", <https://revs.runtime-revolution.com/integrating-a-third-party-api-with-rails-5-134f960d> (12. 12. 2018.)
- Kadoić N., Begičević Redep N., Divjak B., (2017) "Structuring e-Learning Multi-Criteria Decision Making Problems," in Proceedings of 40th Jubilee International Convention, MIPRO 2017, pp. 811–817. <https://doi.org/10.23919/MIPRO.2017.7973514>
- Kadoić N., Divjak B., Begičević Redep N., (2017) "Effective Strategic Decision Making on Open and Distance Education Issues," in Diversity Matters!, pp. 224–234.
- Maheedharan, M., (2017) "How to Pick the Right Web Technology Stack for Your Product", <https://hackernoon.com/how-to-pick-the-right-web-technology-stack-for-your-product-f6d94440af2f> (24. 11. 2018.)
- Gligora Marković, M., Kadoić, N., Kovačić B., (2018) "Selection and prioritization of adaptivity criteria in intelligent and adaptive hypermedia e-learning systems," TEM Journal, 7(1).
- Ochoa, J., (2018) "Consuming third party APIs with Laravel Resources", <https://medium.com/@jeffochoa/consuming-third-pary-apis-with-laravel-resources-c13a0c7dc945> (9.12. 2018.)
- Pankaj, A., (2014) "Spring MVC Internationalization (i18n) and Localization (L10n)", <https://www.journaldev.com/2610/spring-mvc-internationalization-i18n-and-localization-l10n-example> (4. 12. 2018.)
- Raible, M., (2010a, "How I Calculated Ratings for My JVM Web Frameworks Comparison", [https://raibledesigns.com/rd/entry/how\\_i\\_calculated\\_ratings\\_for](https://raibledesigns.com/rd/entry/how_i_calculated_ratings_for) (21. 11. 2018.)
- Raible, M., (2010b) "JVM Web Frameworks Rating Logic", [https://docs.google.com/document/pub?id=1X\\_XvpJd6TgEAMe4a6xxzJ38yzmthvrA6wD7zGy2lgog](https://docs.google.com/document/pub?id=1X_XvpJd6TgEAMe4a6xxzJ38yzmthvrA6wD7zGy2lgog) (22. 11. 2018.)
- Saaty T. L., (2001) "Decision Making with Dependence and Feedback: The Analytic Network Process : the Organization and Prioritization of Complexity", New York: Rws Publications, 2001.
- Stein, D., (2017) "Modern Django—Part 2: REST APIs, Apps, and Django REST Framework", <https://medium.com/@dstein/modern-django-part-2-rest-apis-apps-and-django-rest-framework-ea0cac5ab104> (12. 12. 2018.)
- Technopedia (2018) "Validation", <https://www.techopedia.com/definition/10283/data-validation> (6. 12. 2018.)
- Tomassetti, G., (2018) "A Guide to Code Generation", <https://tomassetti.me/code-generation/> (29. 11. 2018.)
- Tripathi, S. K., (2018) "Top 7 backend web development frameworks in 2018", <https://www.kellontech.com/kellontech-blog/top-7-backend-web-development-frameworks-2018> (28. 11. 2018.)
- Varisco, C., (2016) "How to choose a framework", <https://hackernoon.com/how-to-choose-a-FW-ea8b5b1e1f44> (23. 11. 2018.)
- <https://www.dice.com/> (30. 11. 2018.)
- <https://www.djangoproject.com/> (29. 11. 2018.)

<https://docs.djangoproject.com/en/2.1/topics/templates/> (1. 12. 2018.)  
<https://docs.djangoproject.com/en/2.1/topics/i18n/> (4. 12. 2018.)  
<https://docs.djangoproject.com/en/2.1/topics/testing/> (6. 12. 2018.)  
<https://docs.djangoproject.com/en/2.1/ref/validators/> (6. 12. 2018.)  
<https://docs.spring.io/spring/docs/current/spring-framework-reference/testing.html> (6. 12. 2018.)  
<https://docs.spring.io/spring/docs/4.1.x/spring-framework-reference/html/validation.html> (8. 12. 2018.)  
<https://docs.spring.io/spring/docs/current/spring-framework-reference/languages.html#kotlin> (13. 12. 2018.)  
<https://genius-project.io/en/> (29. 11. 2018.)  
<https://github.com/topics/framework> (27. 11. 2018.)  
<https://github.com/CrestApps/laravel-code-generator> (29. 11. 2018.)  
<https://github.com/jumilla/laravel-extension> (13. 12. 2018.)  
<https://guides.rubyonrails.org/generators.html> (29. 11. 2018.)  
[https://guides.rubyonrails.org/rails\\_application\\_templates.html](https://guides.rubyonrails.org/rails_application_templates.html) (1. 12. 2018.)  
<https://guides.rubyonrails.org/i18n.html> (4. 12. 2018.)  
<https://guides.rubyonrails.org/testing.html> (6. 12. 2018.)  
[https://guides.rubyonrails.org/active\\_record\\_validations.html](https://guides.rubyonrails.org/active_record_validations.html) (6. 12. 2018.)  
<https://guides.rubyonrails.org/v3.2.8/plugins.html> (13. 12. 2018.)  
<https://hotframeworks.com> (28. 11. 2018.)  
<https://laracasts.com/> (29. 11. 2018.)  
<https://laravel.com/docs/5.7/blade> (29. 12. 2018.)  
<https://laravel.com/docs/5.7/localization> (4. 12. 2018.)  
<https://laravel.com/docs/5.7/testing> (6. 12. 2018.)  
<https://laravel.com/docs/5.7/validation> (6. 12. 2018.)  
<https://mit-license.org/> (8. 12. 2018.)  
<https://pypi.org/project/django-plugins/> (13. 12. 2018.)  
<https://spring.io/> (29. 11. 2018.)  
<https://spring.io/understanding/view-templates> (1. 12. 2018.)  
<https://spring.io/projects/spring-integration> (13. 12. 2018.) [https://doi.org/10.1016/S1350-4789\(18\)30494-X](https://doi.org/10.1016/S1350-4789(18)30494-X)  
<https://stackshare.io/stackups/django-vs-laravel-vs-rails> (15. 12. 2018)  
<https://stackshare.io/stackups/django-vs-rails-vs-spring> (15. 12. 2018)  
<https://symfony.com/ten-criteria> (23. 11. 2018.)  
[https://trends.google.com/trends/explore?date=today%205-y&q=%2Fm%2F0jwy148,%2Fm%2F0505cl,%2Fm%2F0d5b,%2Fm%2F06y\\_qx](https://trends.google.com/trends/explore?date=today%205-y&q=%2Fm%2F0jwy148,%2Fm%2F0505cl,%2Fm%2F0d5b,%2Fm%2F06y_qx) (14. 12. 2018.)  
<https://www.guru99.com/application-testing.html> (6. 12. 2018.)  
<http://www.linfo.org/bsdlicense.html> (8. 12. 2018.)  
<https://www.linkedin.com/feed/> (29. 11. 2018.)  
[https://www.slant.co/versus/1233/1746/~ruby-on-rails\\_vs\\_django](https://www.slant.co/versus/1233/1746/~ruby-on-rails_vs_django) (15. 12. 2018)

Stručni rad

<https://doi.org/10.31784/zvr.7.1.10>

Datum primitka: 15. 1. 2019.

Datum prihvatanja: 7. 3. 2019

## USPOREDBA BACK-END FRAMEWORKA ZA RAZVOJ WEB-APLIKACIJA

**Marin Kaluža**

Dr. sc., v. pred., Veleučilište u Rijeci, Vukovarska 58, 51 000 Rijeka, Hrvatska; e-mail: mkaluza@veleri.hr

**Marijana Kalanj**

Studentica, Veleučilište u Rijeci, Vukovarska 58, 51 000 Rijeka, Hrvatska; e-mail: mkalanj@veleri.hr

**Bernard Vukelić**

Dr. sc., v. pred., Veleučilište u Rijeci, Vukovarska 58, 51 000 Rijeka, Hrvatska; e-mail: bvukelic@veleri.hr

### SAŽETAK

Današnje web-aplikacije imaju složenu programsku strukturu, pa se za djelotvornije i brže pisanje programskog koda često koriste frameworkovi. Na tržištu postoje brojni frameworkovi koji se razlikuju po raznim karakteristikama i koriste za izgradnju različitih dijelova arhitekture softvera. U radu se promatraju samo back-end frameworkovi za izradu web-aplikacija. Pokazana su prethodna istraživanja o kriterijima za ocjenjivanje back-end frameworkova i razni izvori koji prema različitim razlozima prikazuju rang-liste popularnosti back-end frameworkova. Opisan je postupak odabira kriterija za usporedbu frameworkova i razlozi odabira frameworkova za analizu (Laravel, Rails, Django i Spring). Izvršeno je ispitivanje (provjera) i ocjenjivanje stupnja zadovoljavanja odabranih osnovnih i dodatnih kriterija usporedbe na odabranim frameworkovima. Analiza pokazuje neznačajne različitosti u ukupnim ocjenama analiziranih frameworkova prema osnovnim kriterijima usporedbe. Prema dodatnim kriterijima usporedbe analiza pokazuje značajnije različitosti u ukupnim ocjenama analiziranih frameworkova. Pokazuje se da je framework Django, zbog toga što je prikladan za razvoj većih aplikacija, prikladan za developere početnike i da ostvaruje najveću ukupnu ocjenu dodatnih kriterija usporedbe, a za developera predstavlja najbolji izbor između analiziranih frameworkova pri razvoju web-aplikacija.

**Ključne riječi:** framework, back-end, kriteriji usporedbe