

SafariBookings

Test Assignment Senior Developer

budget : Max 40 hours

Contents	Page
Intro	2
Developer requirements	3
Functional requirements	4
Technical requirements	8

Version 1.0
Martijn Segers
2023-01-31

Intro

SafariBookings is the world's largest resource and leading authority for planning an African safari. Our online marketplace enables travelers to compare safari offers from top rated tour operators.

This test assignment has a time budget of 40 hours maximum and will be used to determine the level, working method and attitude of the developer.

After approximately half-way, or 20 hours, an evaluation moment should be scheduled, initiated by the developer. After a maximum of 40 hours, what has been developed must be delivered and will be assessed by SafariBookings.

SafariBookings values quality over quantity. We expect the developer to display a proactive attitude. For questions and ambiguities, we will assign one of our senior developers.

Developer requirements

To be able to do this test assignment, we expect the developer to have :

Skill set

- Advanced knowledge of Laravel PHP development (MVC), HTML, CSS and JavaScript
- Good knowledge of MySQL database design
- Knowledge of GIT version control
- Proactive team player attitude

Tools needed

- IDE for PHP development
- (Local) Web hosting environment
- MySQL Database

Also .. provided by us:

- Development contact for questions and ambiguities
- GIT access to the test assignment Laravel project

Documentation for this test assignment:

See GIT (*Laravel Project Root/storage/docs*)

- Test Assignment Document (this document)
- Coding Style Guide
- Link to design
- Basic setup instructions for the Laravel project
- SQL file with the 'operators' and 'accommodations' tables (not needed if you follow the basic setup instructions)

Functional requirements

The functional requirements consist of the following parts:

- Description of the functionality
- The different functional parts
- The graphic design for the views

Description of the functionality

Within the playing field where SafariBookings is active, there are tour operators and accommodations that have provided us with an external link to their website.

It is important to us that the validity of these links is tested periodically so that we have an overview of which external links are no longer valid, so that we can follow up on this.

If there are functional questions, be proactive and contact the SafariBookings development contact to get the answers.

The different functional parts

The system to be developed should include the following two components:

- **The link scanning component**

A Laravel console command that will periodically check the status of the external links and save the results into the appropriate database tables.

If the result is a redirect (301, 302 etc.), we also want to store the destination of the redirect.

- **The 'invalid links' view**

This view should display a list with the invalid links sorted on duration. The list should always have paging navigation and display the following columns:

- **Type**

- The type of the external link (tour operator or accommodation)

- **Name/ External link**

- the operator or accommodation name accompanied by the tested link (the link should point to the actual link in a new tab)

- **Duration**

- The duration in days that the link is invalid.

- **Status**

- The latest invalid status of this link (400, 404, 301, etc.)

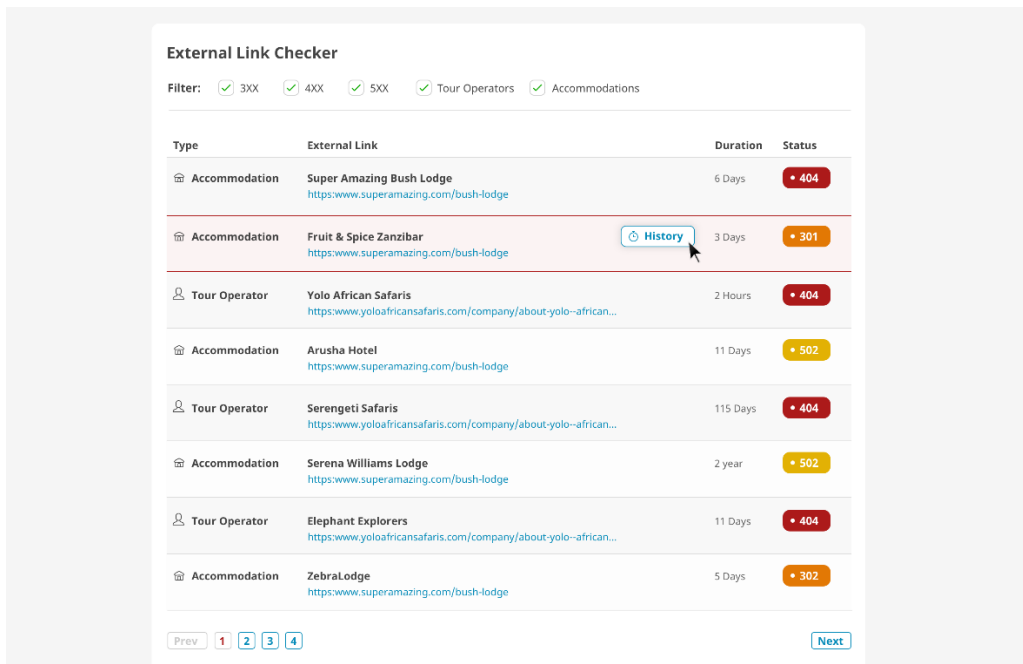
It should be possible to filter the list on : 3xx, 4xx, 5xx error code, tour operators and accommodations.

When hovering over an external link row, a history button is displayed. Clicking this button should display an ‘overlaid’ (centered) ‘History’ view with the link’s status history in a list displaying:

- **Date**
Status datetime
- **Status**
Status code

The graphic designs for the views

The designs for the views are provided in separate png files (see {Laravel project}/storage/docs) , and need to be implemented as pixel-precise as possible.



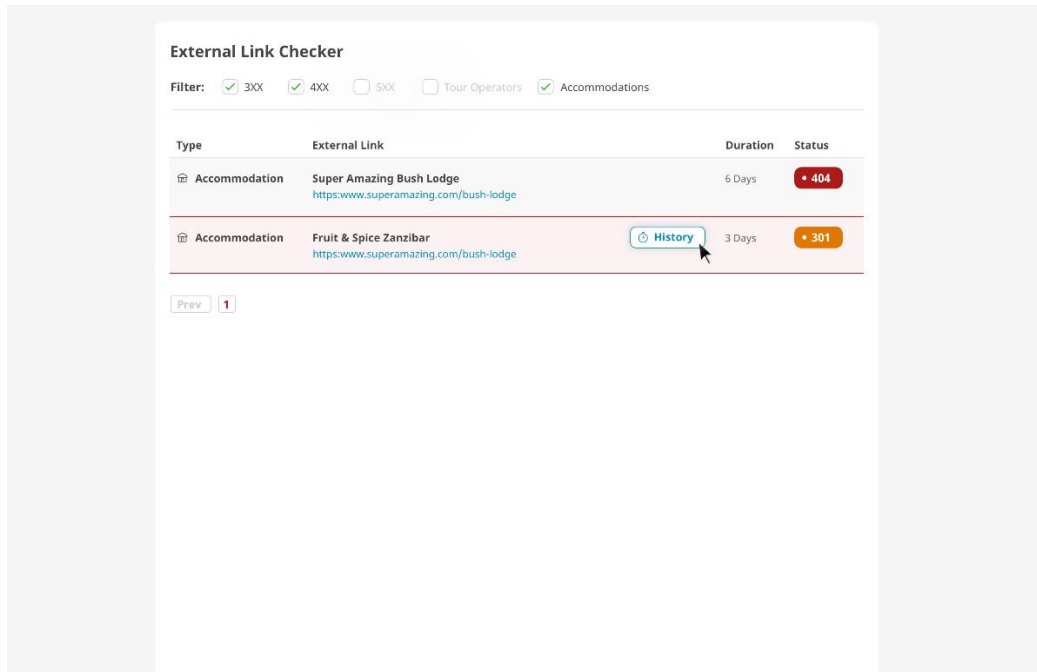
External Link Checker

Filter: ☒ 3XX ☒ 4XX ☒ 5XX ☒ Tour Operators ☒ Accommodations

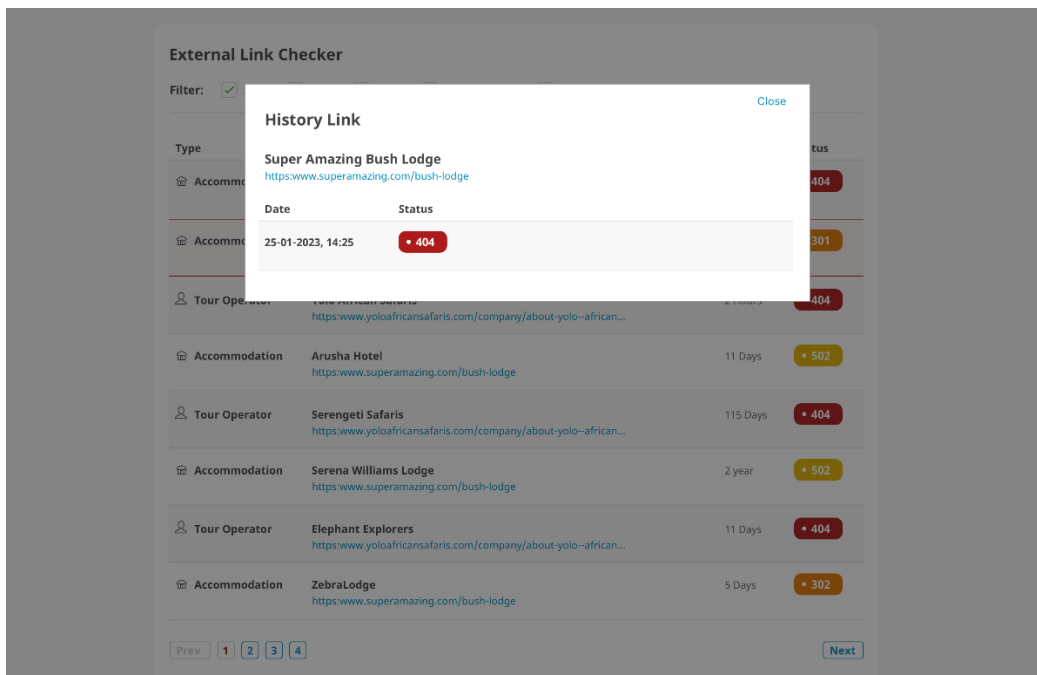
Type	External Link	Duration	Status
Accommodation	Super Amazing Bush Lodge https://www.superamazing.com/bush-lodge	6 Days	404
Accommodation	Fruit & Spice Zanzibar https://www.superamazing.com/bush-lodge	3 Days	301
Tour Operator	Yolo African Safaris https://www.yoloafricansafaris.com/company/about-yolo-african...	2 Hours	404
Accommodation	Arusha Hotel https://www.superamazing.com/bush-lodge	11 Days	502
Tour Operator	Serengeti Safaris https://www.yoloafricansafaris.com/company/about-yolo-african...	115 Days	404
Accommodation	Serena Williams Lodge https://www.superamazing.com/bush-lodge	2 year	502
Tour Operator	Elephant Explorers https://www.yoloafricansafaris.com/company/about-yolo-african...	11 Days	404
Accommodation	ZebraLodge https://www.superamazing.com/bush-lodge	5 Days	302

Prev 1 2 3 4 Next

Main view of the failed extern link list



Filtered view of the failed extern link list



'History' view of a specific external link

Technical requirements

The technical requirements consist of the following parts:

- General technical requirements
- Technical requirements of the functional parts

General technical requirements

- This project has to be coded in PHP Laravel using the model-view-controller (MVC) architecture
- A SafariBookings 'Coding Style Guide' has been provided for coding guidelines
- A MySQL database has to be used for this project. SQL Files for the operators and accommodation tables will be provided
- The CSS file for this project has to be created with the Sass pre-processor
- The designs for the views are provided in separate .png files and need to be implemented as pixel-precise as possible
- Work-in-progress code should be committed to the Git repository after each coding session so that the progress can be followed

If there are technical questions, be proactive and contact the SafariBookings development contact to get your answers.

Technical requirements of the functional parts

The link scanning component

- The scanning component should be built as a Laravel console command
- The scanning component should have a Laravel config (configuration) containing the following settings:
 - maximum number of links to be scanned in one scan run
 - scanning frequency of a link
 - duration after which an external link with a faulty status will be marked as invalid
- The links must be collected from the operators and accommodation tables
- A database table should be created to store the external links that will be scanned
- A database table should be created to store the status history of a specific external link
- When the access of the external link results in a redirect status like 301, the redirect destination needs to be stored into the database as well

The 'invalid links' view

- The (page-) data, or row-html, for the list should be retrieved using Ajax
- The jQuery JavaScript framework may be used
- The window location URL needs to be adjusted for each page and added to the browser's history to support the back button
- The invalid links list should have a config (configuration) that contains the following settings:
 - Number of rows to display per page