Request Handling & Responses

Maximilian Schwarzmüller

AUTHOR

@maxedapps www.mschwarzmueller.com
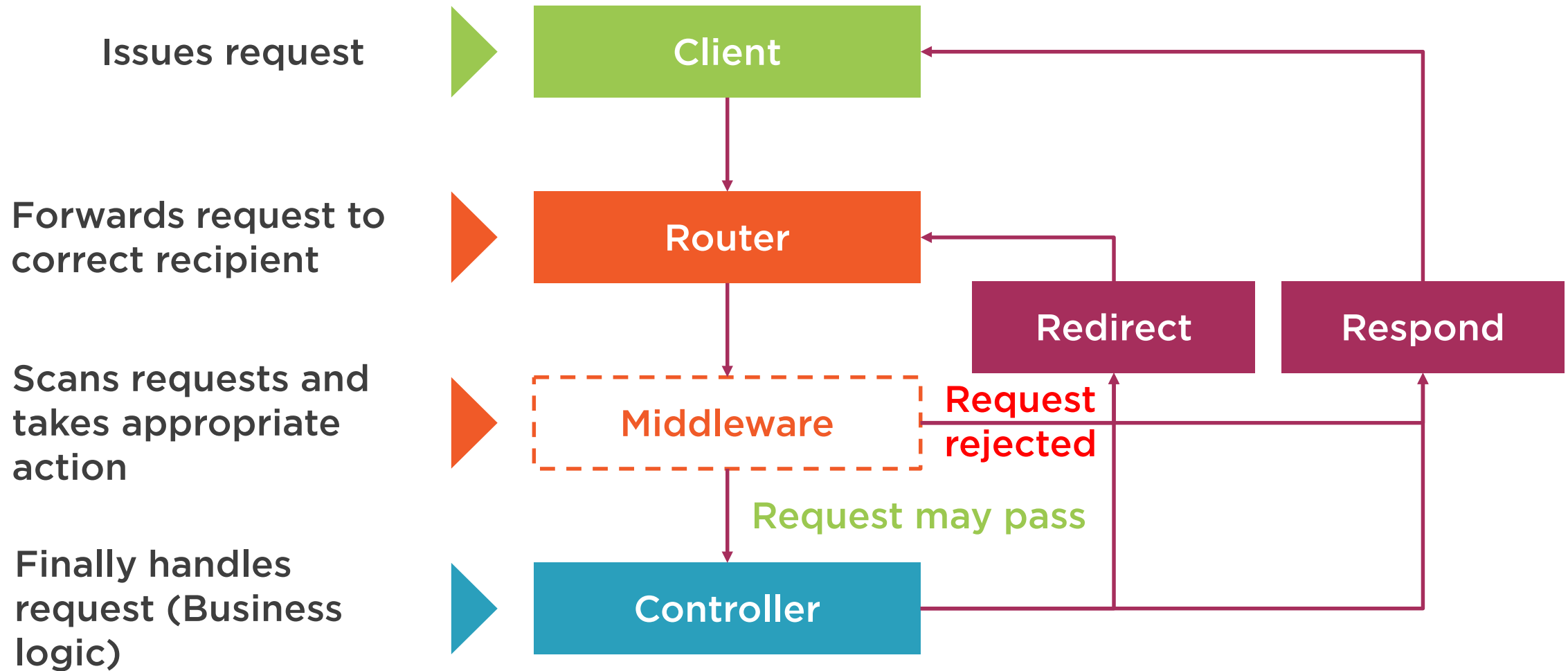
Overview

What to do with requests?

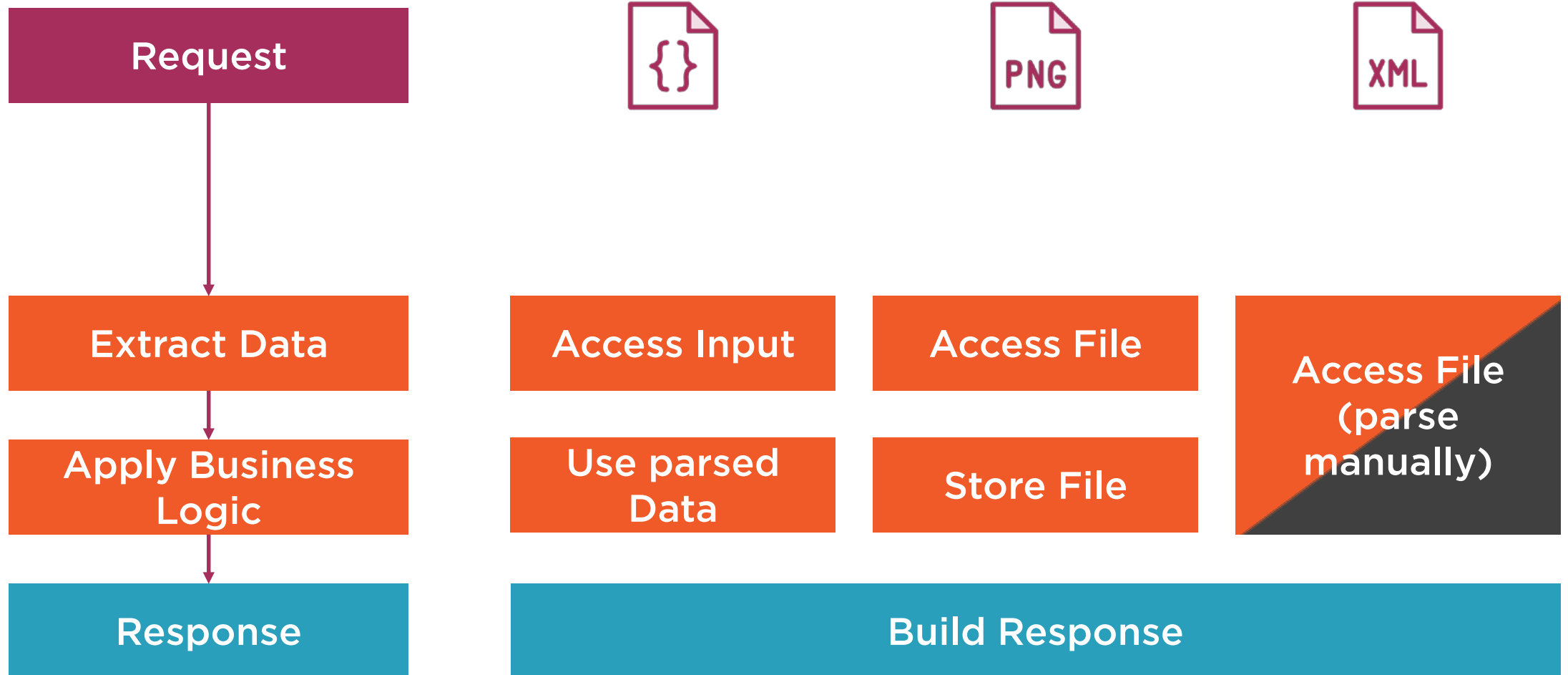Responding appropriately

A case for input validation

Simplified Laravel Request & Response Flow

Issues request ▶ **Client**

Forwards request to correct recipient ▶ **Router**

Redirect   **Respond**

Scans requests and takes appropriate action ▶ Middleware

Request rejected

Request may pass

Finally handles request (Business logic) ▶ **Controller**

Handling Request Input

Planned Controller Duties

Request	{}	PNG	XML
Extract Data	Access Input	Access File	Access File (parse manually)
Apply Business Logic	Use parsed Data	Store File	
Response	Build Response		

Accessing Input

```php
// JSON/ urlencoded data

$title = $request->input('title');


// File

$photo = $request->file('photo');
```

Responses

Common Responses

HTML Page

JSON

File

Plain Text

Building Responses

```php
// HTML Page / View

return response()->view('index');

// JSON

return response()->json([
    'msg' => 'post created',
    'post' => [
        'title' => $post->title
    ]
], 200);

// File download

return response()->download($pathToFile);
```

Response Content

Message
Success or failure message

Data
Optionally attach retrieved data

Link
Optionally attach link to related resources

Validation

User Profiles

"The Geek"
Provides input exactly as required

"The Hacker"
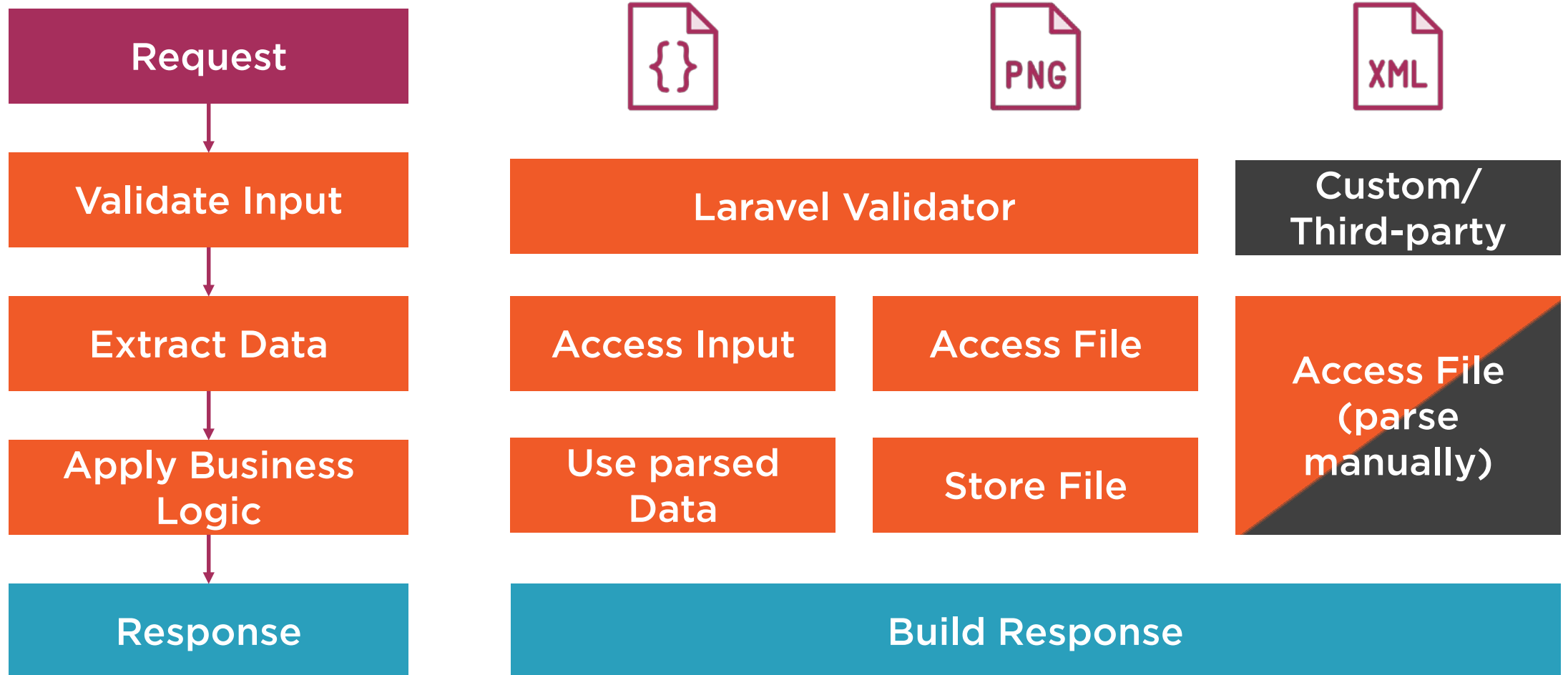Tests your application and its capabilities

"The Normal One"
Forgets required data or data format from time to time

Real Controller Duties

Request	{}	PNG	XML
Validate Input	**Laravel Validator**		**Custom/ Third-party**
Extract Data	**Access Input**	**Access File**	**Access File (parse manually)**
Apply Business Logic	**Use parsed Data**	**Store File**	
Response	**Build Response**		

Input Validation

```php
$this->validate($request, [
    'title' => 'required|max:120',
    'content' => 'required'
]);


// Check filetype

$this->validate($request, [
    'photo' => 'mimes:jpg,png'
]);
```