

Digitaless Museum Backend

Version 0.0.1

Paths List all operations

/person

GET /person

Description

Returns all relevant Person information

Responses

Code	Description	Schema
200	Success	<div>↔</div> <div>▼MultiplePersonResponse { persons: ▶[] }</div>
default	Error	<div>↔</div> <div>▼ErrorResponse { message: string * }</div>

Try this operation

/person/new

Code

Description

PUT /person/new

Description

Add a new Person

Parameters

Name	Located in	Description	Required	Schema
person	body	Person to add	Yes	<div><div>↔</div><div><div>▼PersonResponse {</div><div>firstname: string *</div><div>lastname: string *</div><div>caption: string *</div><div>portrait: ▶Image { }</div><div>chips: ▶[]</div><div>dataTiles: ▶[]</div><div>imageTiles: ▶[]</div><div>id: number</div></div></div>

Responses

Code	Description	Schema
200	Success	<div><div>↔</div><div><div>▼CreatedResponse {</div><div>id: number *</div></div></div>

Try this operation

/person/purgeAll

Code

Description

GET /person/purgeAll

Description

Purge All person

Responses

Code	Description	Schema
200	Success	<div><div>↔</div><div>string</div></div>
default	Error	<div><div>↔</div><div><div>▼ErrorResponse {</div><div>message: string *</div></div></div>

Try this operation

/person/{id}

Name	Located in	Description	Schema	Required	Schema										
GET /person/{id}															
<h3>Description</h3> <p>Get Specific Person</p>															
<h3>Parameters</h3> <table><thead><tr><th>Name</th><th>Located in</th><th>Description</th><th>Required</th><th>Schema</th></tr></thead><tbody><tr><td>id</td><td>path</td><td>ID of Person to fetch</td><td>Yes</td><td>⇔ integer (int64)</td></tr></tbody></table>						Name	Located in	Description	Required	Schema	id	path	ID of Person to fetch	Yes	⇔ integer (int64)
Name	Located in	Description	Required	Schema											
id	path	ID of Person to fetch	Yes	⇔ integer (int64)											
<h3>Responses</h3> <table><thead><tr><th>Code</th><th>Description</th><th>Schema</th></tr></thead><tbody><tr><td>200</td><td>Success</td><td>⇔ <div>▼PersonResponse { firstname: string * lastname: string * caption: string * portrait: ►Image { } chips: ►[] dataTiles: ►[] imageTiles: ►[] id: number }</div></td></tr><tr><td>default</td><td>Error</td><td>⇔ <div>▼ErrorResponse { message: string * }</div></td></tr></tbody></table>						Code	Description	Schema	200	Success	⇔ <div>▼PersonResponse { firstname: string * lastname: string * caption: string * portrait: ►Image { } chips: ►[] dataTiles: ►[] imageTiles: ►[] id: number }</div>	default	Error	⇔ <div>▼ErrorResponse { message: string * }</div>	
Code	Description	Schema													
200	Success	⇔ <div>▼PersonResponse { firstname: string * lastname: string * caption: string * portrait: ►Image { } chips: ►[] dataTiles: ►[] imageTiles: ►[] id: number }</div>													
default	Error	⇔ <div>▼ErrorResponse { message: string * }</div>													

Try this operation

Code	Located in	Description	Schema	Required	Schema
PATCH	/person/{id}	Update Specific Person			
Description					
Update Specific Person					
Parameters					
Name	Located in	Description		Required	Schema
id	path	ID of Person to update		Yes	⇔ integer (int64)
person body		Person Information, or parts thereof	Yes	⇔	<div>▼PersonPatch { firstname: string lastname: string portrait: ▶Image { } chips: ▶[] dataTiles: ▶[] imageTiles: ▶[] id: number }</div>
Responses					
Code		Description			Schema
200		Success			⇔ string
default		Error			<div>▼ErrorResponse { message: string * }</div>
Try this operation					

DELETE

/person/{id}

Name

Located in

Description

Schema

Required

Schema

Description

Delete Specific Person

Parameters

Name	Located in	Description	Required	Schema
id	path	ID of Person to delete	Yes	⇌ integer (int64)

Responses

Code	Description	Schema
200	Success	⇌ string
default	Error	⇌ <div>▼ErrorResponse { message: string * }</div>

Try this operation

Models

ErrorResponse

```
▼ErrorResponse {  
  ⇌ message: string *  
}
```

MultiplePersonResponse

```
▼MultiplePersonResponse {  
  ⇌ persons: ▶[]  
}
```

PersonPatch

```
▼PersonPatch {  
  firstname: string  
  lastname: string  
  portrait: ▶Image { }  
  ⇌ chips: ▶[]  
  dataTiles: ▶[]  
  imageTiles: ▶[]  
  id: number  
}
```

PersonResponse

```

    ▼PersonResponse {
      firstname: string *
      lastname: string *
      caption: string *
      portrait: ▶Image { }
    ⇒ chips: ▶[]
      dataTiles: ▶[]
      imageTiles: ▶[]
      id: number
    }

```

Image

```

    ▼Image {
      url: string *
      width: number
    ⇒ height: number
      source: string
      caption: string
    }

```

Chips

```

    ▼Chips {
    ⇒ letter: string *
      text: string *
    }

```

DataTile

```

    ▼DataTile {
      button_text: string *
    ⇒ long_text: string
      short_text: string *
    }

```

CreatedResponse

```

    ▼CreatedResponse {
    ⇒ id: number *
    }

```