

API Reference

RecipEase.Server

API Version: 1.0

INDEX

1. WF	3
1.1 GET /api/WF	3
1.2 POST /api/WF	3
1.3 GET /api/WF/{id}	5
1.4 PUT /api/WF/{id}	5
1.5 DELETE /api/WF/{id}	6
2. WEATHERFORECAST	7
2.1 GET /api/WeatherForecast	7
2.2 POST /api/WeatherForecast	7
2.3 GET /api/WeatherForecast/{id}	8
2.4 PUT /api/WeatherForecast/{id}	9
2.5 DELETE /api/WeatherForecast/{id}	10

API

1. WF

1.1 GET /api/WF

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: Success

RESPONSE MODEL - text/plain

```
[{  
  Array of object:  
    id            integer  
    date*         string  
    temperatureC  integer  
    summary       string   max:20 chars  
    temperatureF  integer  READ-ONLY  
    humidity      number  
}]
```

RESPONSE MODEL - application/json

```
[{  
  Array of object:  
    id            integer  
    date*         string  
    temperatureC  integer  
    summary       string   max:20 chars  
    temperatureF  integer  READ-ONLY  
    humidity      number  
}]
```

RESPONSE MODEL - text/json

```
[{  
  Array of object:  
    id            integer  
    date*         string  
    temperatureC  integer  
    summary       string   max:20 chars  
    temperatureF  integer  READ-ONLY  
    humidity      number  
}]
```

1.2 POST /api/WF

REQUEST

REQUEST BODY - application/json

```
{  
  id            integer
```

```

date*      string
temperatureC integer
summary    string  max:20 chars
temperatureF integer READ-ONLY
humidity   number
}

```

REQUEST BODY - text/json

```

{
  id          integer
  date*       string
  temperatureC integer
  summary     string  max:20 chars
  temperatureF integer READ-ONLY
  humidity    number
}

```

REQUEST BODY - application/*+json

```

{
  id          integer
  date*       string
  temperatureC integer
  summary     string  max:20 chars
  temperatureF integer READ-ONLY
  humidity    number
}

```

RESPONSE

STATUS CODE - 200: Success

RESPONSE MODEL - text/plain

```

{
  id          integer
  date*       string
  temperatureC integer
  summary     string  max:20 chars
  temperatureF integer READ-ONLY
  humidity    number
}

```

RESPONSE MODEL - application/json

```

{
  id          integer
  date*       string
  temperatureC integer
  summary     string  max:20 chars
  temperatureF integer READ-ONLY
  humidity    number
}

```

RESPONSE MODEL - text/json

```

{
  id          integer
  date*       string
  temperatureC integer
  summary     string  max:20 chars
  temperatureF integer READ-ONLY
}

```

```
    humidity    number
  }
```

1.3 GET /api/WF/{id}

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int32	

RESPONSE

STATUS CODE - 200: Success

RESPONSE MODEL - text/plain

```
{
  id            integer
  date*         string
  temperatureC  integer
  summary       string   max:20 chars
  temperatureF  integer  READ-ONLY
  humidity      number
}
```

RESPONSE MODEL - application/json

```
{
  id            integer
  date*         string
  temperatureC  integer
  summary       string   max:20 chars
  temperatureF  integer  READ-ONLY
  humidity      number
}
```

RESPONSE MODEL - text/json

```
{
  id            integer
  date*         string
  temperatureC  integer
  summary       string   max:20 chars
  temperatureF  integer  READ-ONLY
  humidity      number
}
```

1.4 PUT /api/WF/{id}

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int32	

REQUEST BODY - application/json

```
{
  id            integer
  date*         string
  temperatureC  integer
  summary       string   max:20 chars
  temperatureF  integer  READ-ONLY
  humidity      number
}
```

REQUEST BODY - text/json

```
{
  id            integer
  date*         string
  temperatureC  integer
  summary       string   max:20 chars
  temperatureF  integer  READ-ONLY
  humidity      number
}
```

REQUEST BODY - application/*+json

```
{
  id            integer
  date*         string
  temperatureC  integer
  summary       string   max:20 chars
  temperatureF  integer  READ-ONLY
  humidity      number
}
```

RESPONSE

STATUS CODE - 200: Success

1.5 DELETE /api/WF/{id}

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int32	

RESPONSE

STATUS CODE - 200: Success

2. WEATHERFORECAST

2.1 GET /api/WeatherForecast

Returns all WeatherForecasts.

Retrieves all items and all attributes from the `weatherforecasts` table.

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: Success

RESPONSE MODEL - application/json

```
[ {  
  Array of object:  
    id            integer  
    date*         string  
    temperatureC  integer  
    summary       string  max:20 chars  
    temperatureF  integer  READ-ONLY  
    humidity      number  
  } ]
```

2.2 POST /api/WeatherForecast

Adds the new forecast to the database.

Leave `id` out or specify a value of 0 to have it be automatically generated.

REQUEST

REQUEST BODY - application/json

```
{  
  id            integer  
  date*         string  
  temperatureC  integer  
  summary       string  max:20 chars  
  temperatureF  integer  READ-ONLY  
  humidity      number  
}
```

RESPONSE

STATUS CODE - 201: Success

RESPONSE MODEL - application/json

```
{  
  id            integer  
  date*         string  
  temperatureC  integer  
  summary       string  max:20 chars
```

```
    temperatureF integer READ-ONLY
    humidity      number
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  type      string
  title     string
  status    integer
  detail    string
  instance  string
}
```

STATUS CODE - default: Error

RESPONSE MODEL - application/json

```
{
  type      string
  title     string
  status    integer
  detail    string
  instance  string
}
```

2.3 GET /api/WeatherForecast/{id}

Returns the WeatherForecast with the given id.

Retrieves the item with the given id value in the ID column from the weatherforecasts table.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int32	The ID of the WeatherForecast to retrieve.

RESPONSE

STATUS CODE - 200: Success

RESPONSE MODEL - application/json

```
{
  id          integer
  date*      string
  temperatureC integer
  summary     string  max:20 chars
  temperatureF integer READ-ONLY
  humidity    number
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
```



```

type      string
title     string
status    integer
detail    string
instance  string
}

```

STATUS CODE - default: Error

RESPONSE MODEL - application/json

```

{
  type      string
  title     string
  status    integer
  detail    string
  instance  string
}

```

2.4 PUT /api/WeatherForecast/{id}

Replaces the forecast in the database.

Ensure the `id` in the body matches the id of the request URL, otherwise a 400 response will be returned.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int32	The ID of the WeatherForecast to update.

REQUEST BODY - application/json

```

{
  id          integer
  date*       string
  temperatureC integer
  summary     string  max:20 chars
  temperatureF integer READ-ONLY
  humidity    number
}

```

RESPONSE

STATUS CODE - 204: Success

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```

{
  type      string
  title     string
  status    integer
  detail    string
  instance  string
}

```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  type      string
  title     string
  status    integer
  detail    string
  instance  string
}
```

STATUS CODE - default: Error

RESPONSE MODEL - application/json

```
{
  type      string
  title     string
  status    integer
  detail    string
  instance  string
}
```

2.5 DELETE /api/WeatherForecast/{id}

Removes the forecast from the database.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int32	The ID of the WeatherForecast to delete.

RESPONSE

STATUS CODE - 200: Success

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  type      string
  title     string
  status    integer
  detail    string
  instance  string
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

```
{
  type      string
  title     string
  status    integer
  detail    string
  instance  string
}
```

STATUS CODE - default: Error

RESPONSE MODEL - application/json

```
{  
  type      string  
  title     string  
  status    integer  
  detail    string  
  instance  string  
}
```
