# End-Points de la AP para usar en ApiLaravel

**+ Url base: http://127.0.0.1:8000/api**

**+ Register: /register**

Hay que enviar los siguientes valores:
- ✓ name: nombre
- ✓ email: correo electronico
- ✓ password: contraseña
- ✓ password_confirmation: confirmación de contraseña

❖ **RESULTADO: Si todo va correcto, devuelve un json compuesto por:**

**{"success": true,**

**"message": "User register successfully.",**

**"data": {**

```
name
email
updated_at
created_at
id
```
    **}**
**}**

❖ **En caso de error:**

✘ Email ya en uso

```
{
"success": false,
 "message": "The email has already been taken."
}
```

✘ Contraseña no coinciden

```
{
"success": false,
  "message": "The password field confirmation does not match."}
```

✘ **Falta de campo password_confirmation u otro**

```
{"success": false,
 "message": "The password field confirmation does not match."
 }
```

**+ Users: /users**

-Hay que enviar los siguientes valores:

Headers:
 + Accept: application/json
 + Authorization: Bearer token devuelto en login

❖ **RESUTADO: Si todo es va bien:**

```
{"success": true,
   "message": "Users retrieved successfully.",
```

```
"data": [
   {"id"
      "name"
      "email"
      "email_confirmed"
      "email_verified_at"
      "actived"
      "type"
      "code"
      "deleted

      "created_at"

      "updated_at"

   }
```

× **Si hay error:**

**- Si no está el token o no coincide o ya no es valido**

```
{

   "message": "Unauthenticated."

}
```

**+ User: /user/{userid}**

Hay que enviar los siguientes valores:
UserId: id del usuario a obtener

**Headers:**

**+ Accept: application/json**

**+ Authorization: Bearer token devuelto en login**

❖ **RESULTADO: Si todo va bien**

```
{

"success": true,

 "message": "User retrieved successfully.",

 "data": {

   "id": 43,

   "name": "admin",

   "email": "admin@gmail.com",

   "email_confirmed": 1,

   "email_verified_at": "2023-11-10T12:16:40.000000Z",

   "actived": 1,

   "type": "a",

   "code": 270057,

   "deleted": 0,
```

```
    "created_at": "2023-11-10T11:16:12.000000Z",

    "updated_at": "2023-11-15T18:11:56.000000Z"

  }

}
```

❖ **Si hay algún error:**
× **No hay usuario con ese Id**

```
{

  "success": false,

  "message": "User not found."

}
```

× **Cualquier otro en error en Controller**

```
[

      'success' => false,

      'message' => 'There was an error retrieving the user.'

  ]
```

✓ **Emotions: /emotions**

Hay que enviar los siguientes datos:

**Headers:**

+ **Accept: application/json**

+ **Authorization: Bearer token devuelto en login**

❖ **RESULTADO: Si todo va bien:**

```
{

  "success"

  "message"

  "data": [

    {

      "id"

      "type"

      "name"

      "user_id"

       updated_at

       created_at

    }

}
```

❖ **En caso de error:**

× **Email ya en uso**

```json
{
  "success": false,
  "message": "The email has already been taken."
}
```

× **Contraseña no coinciden**

```json
{
  "success": false,
  "message": "The password field confirmation does not match."
}
```

× **Falta de campo password_confirmation u otro**

```json
{
  "success": false,
  "message": "The password field confirmation does not match."
}
```

✓ **Login: /login**

Hay que enviar los valores

-Email: Correo Electronico

- Password: Contraseña

❖ **RESULTADO: Si todo va correcto obtenemos:**

```json
{
  "success": true,
  "message": "User login successfully.",
  "data": {
    "access_token":
    "token_type":
    "user": {
      "id":
      "name":
      "email":
      "email_confirmed":
      "email_verified_at":
```

```
        "actived":
        "type":
        "code"
        "deleted":
        "created_at":
        "updated_at"
      }
    }
}
```

❖ **En caso de error puede ser varios:**

✗ **Valores no coinciden**

```
{
    "success": false,
    "message": "Invalid credentials"
}
```

✗ **Usuario No Activado**

```
{
    "success": false,
    "message": "User account is not activated"
}
```

✗ **Correo de Usuario no Confirmado**

```
{
    "success": false,
    "message": "Email not confirmed"
}
```

✓ **Users: /users**

**-Hay que enviar los siguientes valores:**

**Headers:**

  **+ Accept: application/json**

  **+ Authorization: Bearer token devuelto en login**

    ❖ **RESUTADO: Si todo es va bien:**

```
{
    "success": true,
    "message": "Users retrieved successfully.",
```

```
    "data": [
      {
        "id"
        "name"
        "email"
        "email_confirmed"
        "email_verified_at"
        "actived"
        "type"
        "code"
        "deleted
        "created_at"
        "updated_at"
      }
```

❖ **Si hay error:**

× **Si no está el token o no coincide o ya no es valido**

```
{
    "message": "Unauthenticated."
}
```

&#10003; **User: /user/{userid}**

Hay que enviar los siguientes valores:
UserId: id del usuario a obtener

**Headers:**

**+ Accept: application/json**

**+ Authorization: Bearer token devuelto en login**

❖ **RESULTADO: Si todo va bien**

```
{
    "success": true,
    "message": "User retrieved successfully.",
    "data": {
        "id"
        "name":
        "email"
        "email_confirmed"
        "email_verified_at"
```

**"actived"**

**"type"**

**"code":**

**"deleted"**

**"created_at"**

**"updated_at"**

    **}**

**}**

❖ **Si hay algun error:**

✗ **No hay usuario con ese Id**

**{**

  **"success": false,**

  **"message": "User not found."**

**}**

✗ **Cualquier otro en error en Controller**

**[**

    **'success' => false,**

    **'message' => 'There was an error retrieving the user.'**

   **]**

❖ **Emotions: /emotions**

Hay que enviar los siguientes datos:
**Headers:**

  **+ Accept: application/json**

  **+ Authorization: Bearer token devuelto en login**

❖ **RESULTADO: Si todo va bien:**

 **{**

  **"success"**

  **"message"**

  **"data": [**

   **{**

     **"id"**

     **"type"**

     **"name"**

     **"user_id"**

     **"description"**

```
        "image"

        "date"

        "deleted"

        "created_at"

        "updated_at"

    },
```

     ✖   **Si hay algun error:**

**- Si el token no es válido**

```
{

    "message": "Unauthenticated."

}
```

**-Si hay un error al recuperar datos**

```
[

        'success' => false,

        'message' => 'There was an error retrieving the emotions.'

    ]
```

    ✓   **Emotion: /getEmotion/{emotionId}**

Hay que enviar los siguientes valores: Hay que enviar los siguientes valores: EmotionId (id de la emocion a obtener)

**Headers:**

 **+ Accept: application/json**

 **+ Authorization: Bearer token devuelto en login**

    ❖   **RESULTADO: Si todo va bien:**

```
{

    "success": true,

    "message": "Emotion retrieved successfully.",

    "data": {

        "id": 1,

        "type": "emotion",

        "name": "Hambre",

        "user_id": 43,

        "description": "",

        "image": "https://mindcare.allsites.es/public/images/alegria.png",

        "date": null,

        "deleted": 0,
```

```json
    "created_at": "2023-11-07T17:35:31.000000Z",
    "updated_at": "2024-01-30T15:48:46.000000Z"
  }
}
```

❖ **Si hay algún error:**

× **Si no hay ninguna emocion con ese id**

```json
{
  "success": false,
   "message": "Emotion not found."
}
```

× **Cualquier error al recupar la emocion:**

```
[
    'success' => false,
    'message' => 'There was an error retrieving the emotion.'
]
```

✓ **AddEmotion: /addEmotion**

Hay que enviar los siguientes valores:

**name (nombre de la emoción; Tristeza, Alegría, Ira, Miedo, Asco)**

❖ **RESULTADO: Si todo va bien:**

```json
{
  "success": true,
  "message": "Emotion created successfully.",
  "data": {
    "type": "emotion",
    "name": "Iran",
    "user_id": 216,
    "description": "",
    "deleted": 0,
    "image": "https://mindcare.allsites.es/public/images/iran.png",
    "updated_at": "2024-01-31T17:18:55.000000Z",
    "created_at": "2024-01-31T17:18:55.000000Z",
    "id": 15
  }
}
```

**Si hay algun error:**

```
{
  "success": false,
  "message": "There was a database error creating the emotion."
}
    [
        'success' => false,
        'message' => 'There was an error creating the emotion.'
    ]
```

- ✓ **UpdateEmotion: /updateEmotion/{emotionId}**

    Hay que pasarle los siguientes valores:
- ✓ **DeleteEmotion: /deleteEmotion/{emotionId}**

Hay que enviar los siguientes valores:
EmocionId (id de la emoción)

**NameNew (nuevo nombre de la emocion)**

**Headers:**

**+ Accept: application/json**

**+ Authorization: Bearer token devuelto en login**

❖ **RESULTADO: Si todo va bien:**

```
{
  "success": true,
  "message": "Emotion updated successfully.",
  "data": {
    "id": 14,
    "type": "emotion",
    "name": "Tristeza",
    "user_id": 43,
    "description": "",
    "image": "https://mindcare.allsites.es/public/images/miedo.png",
    "date": null,
    "deleted": 1,
    "created_at": "2024-01-30T15:31:39.000000Z",
    "updated_at": "2024-01-31T17:06:09.000000Z"
  }
}
```

- ❖ **Si hay algun error:**
  - ✖ **Ninguna emocion con ese Id**

```
{
   "success": false,
   "message": "Emotion not found."
}
```

  - ✖ **Cualquier error al cambiar datos de la emocion (controller)**

```
[
      'success' => false,
      'message' => 'There was an error updating the emotion.'
    ]
```

**EmotionId (id de la emoción a eliminar)**

**Headers:**

+ **Accept: application/json**

+ **Authorization: Bearer token devuelto en login**

- ❖ **RESULTADO: Si todo sale bien:**

```
{
   "success": true,
   "message": "Emotion deleted successfully."
}
```

- ❖ **Si hay errores:**
  - ✖ **Emocion con el id indicado no encontrado**

```
{
   "success": false,
   "message": "Emotion not found."
}
```

  - ✖ **Error en el proceso de borrado (controller)**

```
[
      'success' => false,
      'message' => 'There was an error deleting the emotion.'
]
```