

INFORME API TRABAJO FINAL

Sistema de Gestión de Cursos en Línea

Objetivo: Administrar la creación, inscripción y evaluación de cursos en línea.

Alumno: Ortega Marcos.

Profesor: Cristhian Cantero.

Curso: 6to 2da.

Escuela: E.P.E.T N°20.

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Para la creación de esta API Reset se utilizó Node.js y Express.

Se utilizó el framework de Express para la creación de solicitudes HTTP como GET, POST, DELETE y PUT. También para la creación de los endpoints.

Hay distintos endpoints para las distintas secciones, estos endpoints son:

- Cursos
 - /ListarCursos (Listar todos los cursos)
 - /BuscarCursos/:id (Busca el curso específico por su id)
 - /SubirCurso (Crear curso)
 - /ActualizarCurso/:id (Actualiza el curso específico por su id)
 - /EstadoCurso/:id (Cambia el estado de ACTIVO/INACTIVO del curso por su id)
- Módulos
 - /ListarModulos (Listar todos los módulos)
 - /BuscarModulo/:id (Busca el modulo específico por su id)
 - /SubirModulo (Crear modulo)
 - /ActualizarModulo/:id (Actualiza el modulo específico por su id)
 - /EstadoModulo/:id (Cambia el estado de ACTIVO/INACTIVO del módulo por su id)
 - /AsignarModulo/:moduloId/:cursoId (Asigna el modulo a un curso por sus id específicos)
- Evaluaciones
 - /ListarEvaluaciones (Listar todas las evaluaciones)
 - /BuscarEvaluacion/:id (Busca la evaluación específica por su id)
 - /SubirEvaluacion (Crear Evaluación)
 - /ActualizarEvaluacion/:id (Actualiza la evaluación específica por su id)
 - /EstadoEvaluacion/:id (Cambia el estado de ACTIVO/INACTIVO de la evaluación por su id)
 - /AsignarEvaluacion/:moduloId/:evaluacionId (Asigna la evaluación a un módulo por sus id específico)
- Alumnos
 - /ListarAlumnos (Listar todos los alumnos)
 - /BuscarAlumno/:dni (Busca el alumno específico por su DNI)
 - /SubirAlumno (Crear Alumno)
 - /ActualizarAlumno/:dni (Actualiza el alumno específico por su DNI)
 - /EstadoAlumno/:dni (Cambia el estado de ACTIVO/INACTIVO del alumno por su DNI)
- Profesores
 - /ListarProfesores (Listar todos los profesores)
 - /BuscarProfesor/:dni (Busca el profesor específico por su DNI)
 - /SubirProfesor (Crear Profesor)

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

- /ActualizarProfesor/:dni (Actualiza el profesor específico por su DNI)
- /EstadoProfesor/:dni (Cambia el estado de ACTIVO/INACTIVO del Profesor por su DNI)
- Inscripciones
 - /ListarInscripciones (Listar todas las inscripciones)
 - /BuscarInscripcionesAlumno/:dni (Busca la inscripción del alumno específico por su DNI)
 - /BuscarInscripcionesCurso/:id (Busca las inscripciones del curso específico por su id)

Básicamente el objetivo de esta API es la gestión de cursos en línea. Se utiliza un archivo JSON como “Base de datos”, para poder manejar operaciones como lectura, alta, baja y modificación de datos.

Para tener en cuenta. La permite el cambio de estado de ACTIVO/INACTIVO para los cursos, módulos, evaluaciones, alumnos y profesores, en lugar de eliminarlos.

A continuación, se van a mostrar imágenes de ejemplos de uso de los distintos endpoints.

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.
-Cursos:

Listamos los cursos.

GET ▼ http://localhost:3000/ListarCursos Send ▼ 200 OK 24 ms 2 B Just Now ▼

Params Body Auth Headers 3 Scripts Docs

URL PREVIEW
http://localhost:3000/ListarCursos Copy

QUERY PARAMETERS Import from URL Bulk Edit
+ Add Delete all Description

name	value	▼	☑	🗑
------	-------	---	---	---

PATH PARAMETERS
Path parameters are url path segments that start with a colon ':'. e.g. ':id' ×

Preview ▼

```
1 []
```

\$.store.books[*].author 🔍

Online Made with ♥ by Kong

Creamos un Curso

POST ▼ http://localhost:3000/SubirCurso Send ▼ 200 OK 30 ms 253 B Just Now ▼

Params Body ● Auth Headers 4 Scripts Docs

JSON ▼

```
1 {  
2   "nombre": "Matematicas Basicas",  
3   "descripcion": "Curso introductorio de  
4   matematicas",  
5   "fechaInicio": "2024-01-10",  
6   "fechaFin": "2024-06-10",  
7   "duracion": "6 meses",  
8   "cantidadTotalAlumnos": 35,  
9   "estado": "INACTIVO"  
}
```

Preview ▼

```
1 {  
2   "idCurso": 1,  
3   "nombre": "Matematicas Basicas",  
4   "descripcion": "Curso introductorio de matematicas",  
5   "fechaInicio": "2024-01-10",  
6   "fechaFin": "2024-06-10",  
7   "duracion": "6 meses",  
8   "cantidadTotalAlumnos": 35,  
9   "estado": "INACTIVO",  
10  "cantidadAlumnos": 0,  
11  "cantidadModulos": 0  
12 }
```

Beautify JSON 🔍

\$.store.books[*].author

Online Made with ♥ by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos la creación del curso

GET ▼ http://localhost:3000/ListarCursos Send 200 OK 59 ms 255 B Just Now ▼

Params **Body** Auth Headers 3 Scripts Docs

No Body ▼

Preview ▼

```
1 [
2   {
3     "idCurso": 1,
4     "nombre": "Matematicas Basicas",
5     "descripcion": "Curso introductorio de matematicas",
6     "fechaInicio": "2024-01-10",
7     "fechaFin": "2024-06-10",
8     "duracion": "6 meses",
9     "cantidadTotalAlumnos": 35,
10    "estado": "INACTIVO",
11    "cantidadAlumnos": 0,
12    "cantidadModulos": 0
13  }
14 ]
```

\$.store.books[*].author

Online Made with ♥ by Kong

(Se crea con la cantidad de alumnos y módulos en 0)

Habilitamos el curso por su id

DELETE ▼ http://localhost:3000/EstadoCurso/1 Send 200 OK 27 ms 49 B Just Now ▼

Params **Body** Auth Headers 3 Scripts Docs

No Body ▼

Preview ▼

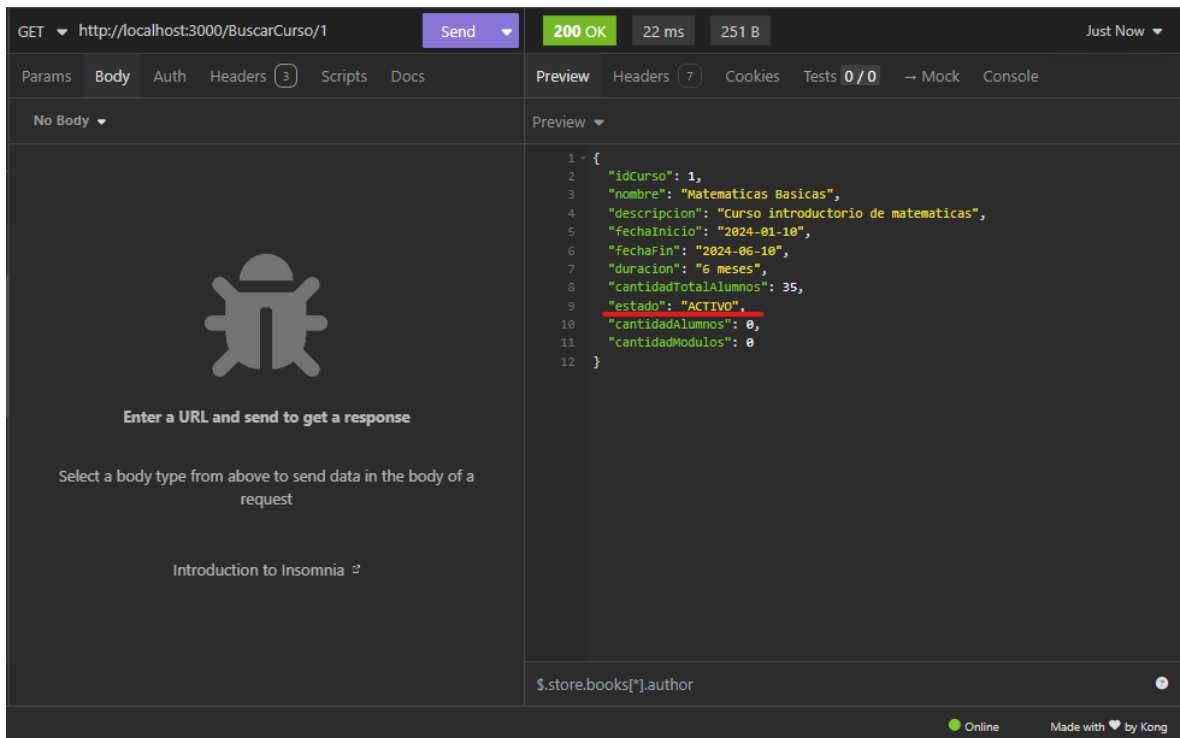
```
1 {
2   "message": "Matematicas Basicas cambio a ACTIVO"
3 }
```

\$.store.books[*].author

Online Made with ♥ by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos el cambio de estado con el id específico del curso



GET ▼ http://localhost:3000/BuscarCurso/1 Send 200 OK 22 ms 251 B Just Now ▼

Params **Body** Auth Headers 3 Scripts Docs


No Body ▼

Preview ▼


```
1 {
2   "idCurso": 1,
3   "nombre": "Matematicas Basicas",
4   "descripcion": "Curso introductorio de matematicas",
5   "fechaInicio": "2024-01-10",
6   "fechaFin": "2024-06-10",
7   "duracion": "6 meses",
8   "cantidadTotalAlumnos": 35,
9   "estado": "ACTIVO",
10  "cantidadAlumnos": 0,
11  "cantidadModulos": 0
12 }
```

Enter a URL and send to get a response

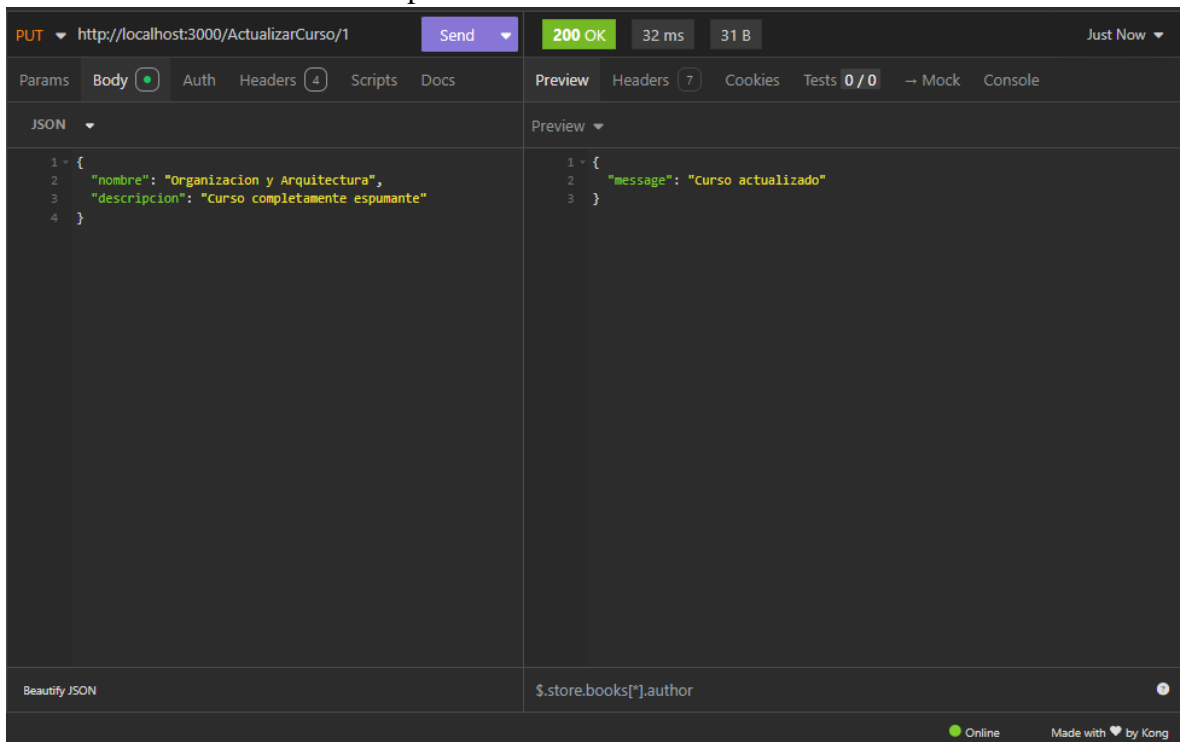
Select a body type from above to send data in the body of a request

Introduction to Insomnia 

\$.store.books[0].author

Online Made with  by Kong

Modificación del curso por id



PUT ▼ http://localhost:3000/ActualizarCurso/1 Send 200 OK 32 ms 31 B Just Now ▼

Params **Body** ● Auth Headers 4 Scripts Docs

JSON ▼


```
1 {
2   "nombre": "Organizacion y Arquitectura",
3   "descripcion": "Curso completamente espumante"
4 }
```

Preview ▼

```
1 {
2   "message": "Curso actualizado"
3 }
```

Beautify JSON

\$.store.books[0].author

Online Made with  by Kong

(En este caso modificamos el nombre y descripción del curso)

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.
Verificamos la modificación.


GET http://localhost:3000/BuscarCurso/1 Send

200 OK 56 ms 254 B Just Now

Params Body Auth Headers 3 Scripts Docs

Preview Headers 7 Cookies Tests 0/0 Mock Console

No Body



Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Introduction to Insomnia

Preview

```
1 {
2   "idCurso": 1,
3   "nombre": "Organizacion y Arquitectura",
4   "descripcion": "Curso completamente espumante",
5   "fechaInicio": "2024-01-10",
6   "fechaFin": "2024-06-10",
7   "duracion": "6 meses",
8   "cantidadTotalAlumnos": 35,
9   "estado": "ACTIVO",
10  "cantidadAlumnos": 0,
11  "cantidadModulos": 0
12 }
```

\$store.books[*].author

Online Made with by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.
-Módulos

Listamos los módulos

GET `http://localhost:3000/ListarModulos` **Send** **200 OK** 22 ms 2 B Just Now

Params Body Auth Headers **3** Scripts Docs

URL PREVIEW
`http://localhost:3000/ListarModulos`

QUERY PARAMETERS
Import from URL Bulk Edit
+ Add Delete all Description

name value

PATH PARAMETERS
Path parameters are url path segments that start with a colon
' e.g. :id'

Preview Headers **7** Cookies Tests **0 / 0** → Mock Console

Preview
1 `[]`

`$.store.books[*].author`

Online Made with ❤ by Kong

Creamos un nuevo modulo

POST `http://localhost:3000/SubirModulo` **Send** **200 OK** 60 ms 144 B Just Now

Params **Body** Auth Headers **4** Scripts Docs

JSON

```
1 {  
2   "nombre": "Algebra",  
3   "duracion": "3 semanas",  
4   "contenido": "Introduccion.",  
5   "cantidadEvaluaciones": 0,  
6   "estado": "INACTIVO"  
7 }
```

Preview Headers **7** Cookies Tests **0 / 0** → Mock Console

Preview
1 {
2 "idModulo": 1,
3 "nombre": "Algebra",
4 "duracion": "3 semanas",
5 "contenido": "Introduccion.",
6 "cantidadEvaluaciones": 0,
7 "estado": "INACTIVO",
8 "idCurso": null
9 }

Beautify JSON `$.store.books[*].author`

Online Made with ❤ by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

(No tiene un curso asignado)

Asignamos el modulo a un curso

The screenshot shows the Insomnia client interface. The top bar indicates a POST request to `http://localhost:3000/AsignarModulo/1/1` with a status of **200 OK**, a response time of **28 ms**, and a size of **81 B**. The left sidebar shows the 'Body' tab selected, with a 'No Body' message and a large grey bug icon. The main area displays the 'Preview' tab, showing the JSON response:

```
1 {  
2   "message": "Algebra añadida correctamente al curso organizacion y  
3   Arquitectura"  
}
```

At the bottom, the JSONPath `$.store.books[*].author` is visible. The footer includes 'Online' status and 'Made with ❤ by Kong'.

Verificamos que si se añadió correctamente por el id especifico del módulo.

The screenshot shows the Insomnia client interface. The top bar indicates a GET request to `http://localhost:3000/BuscarModulo/1` with a status of **200 OK**, a response time of **23 ms**, and a size of **141 B**. The left sidebar shows the 'Body' tab selected, with a 'No Body' message and a large grey bug icon. The main area displays the 'Preview' tab, showing the JSON response:

```
1 {  
2   "idModulo": 1,  
3   "nombre": "Algebra",  
4   "duracion": "3 semanas",  
5   "contenido": "Introduccion.",  
6   "cantidadEvaluaciones": 0,  
7   "estado": "INACTIVO",  
8   "idCurso": 1  
9 }
```

At the bottom, the JSONPath `$.store.books[*].author` is visible. The footer includes 'Online' status and 'Made with ❤ by Kong'.

(El modulo está asignado al curso con id 1)

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos la cantidad de módulos del curso con Id 1.

The screenshot shows the Insomnia client interface. The top bar indicates a GET request to `http://localhost:3000/BuscarCurso/1` with a status of **200 OK**, a response time of **16 ms**, and a size of **254 B**. The left sidebar shows the 'Body' tab selected, with a 'No Body' message and a large bug icon. The main area displays the 'Preview' tab, showing a JSON response with the following structure:

```
1 {
2   "idCurso": 1,
3   "nombre": "Organizacion y Arquitectura",
4   "descripcion": "Curso completamente espumante",
5   "fechaInicio": "2024-01-10",
6   "fechaFin": "2024-06-10",
7   "duracion": "6 meses",
8   "cantidadTotalAlumnos": 35,
9   "estado": "ACTIVO",
10  "cantidadAlumnos": 0,
11  "cantidadModulos": 1
12 }
```

The value `"cantidadModulos": 1` is highlighted with a red underline. At the bottom, there is a status bar with a green 'Online' indicator and the text 'Made with ❤ by Kong'.

Habilitamos el modulo por su id.

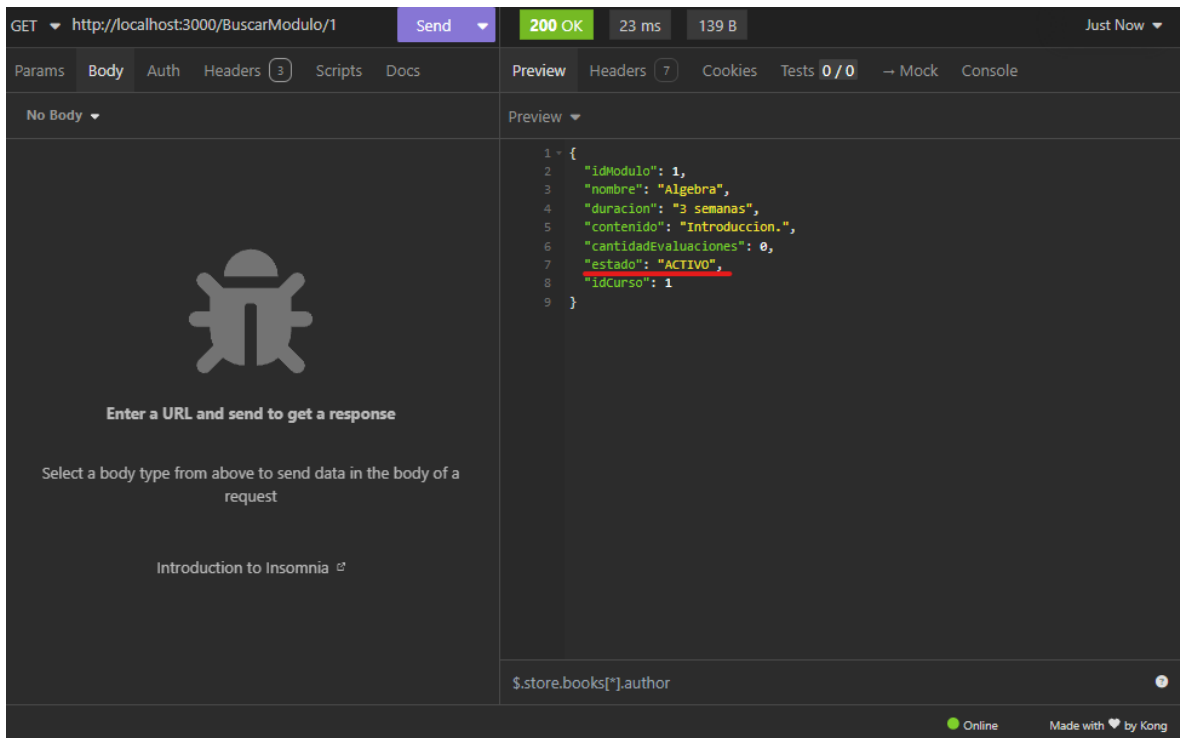
The screenshot shows the Insomnia client interface. The top bar indicates a DELETE request to `http://localhost:3000/EstadoModulo/1` with a status of **200 OK**, a response time of **17 ms**, and a size of **47 B**. The left sidebar shows the 'Body' tab selected, with a 'No Body' message and a large bug icon. The main area displays the 'Preview' tab, showing a JSON response with the following structure:

```
1 {
2   "message": "El modulo Algebra cambio a ACTIVO"
3 }
```

At the bottom, there is a status bar with a green 'Online' indicator and the text 'Made with ❤ by Kong'.

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos el cambio de estado del módulo.



GET ▼ http://localhost:3000/BuscarModulo/1 Send 200 OK 23 ms 139 B Just Now ▼

Params **Body** Auth Headers 3 Scripts Docs **Preview** Headers 7 Cookies Tests 0/0 → Mock Console

No Body ▼

Preview ▼

```
1 {  
2   "idModulo": 1,  
3   "nombre": "Algebra",  
4   "duracion": "3 semanas",  
5   "contenido": "Introduccion.",  
6   "cantidadEvaluaciones": 0,  
7   "estado": "ACTIVO",  
8   "idCurso": 1  
9 }
```

Enter a URL and send to get a response

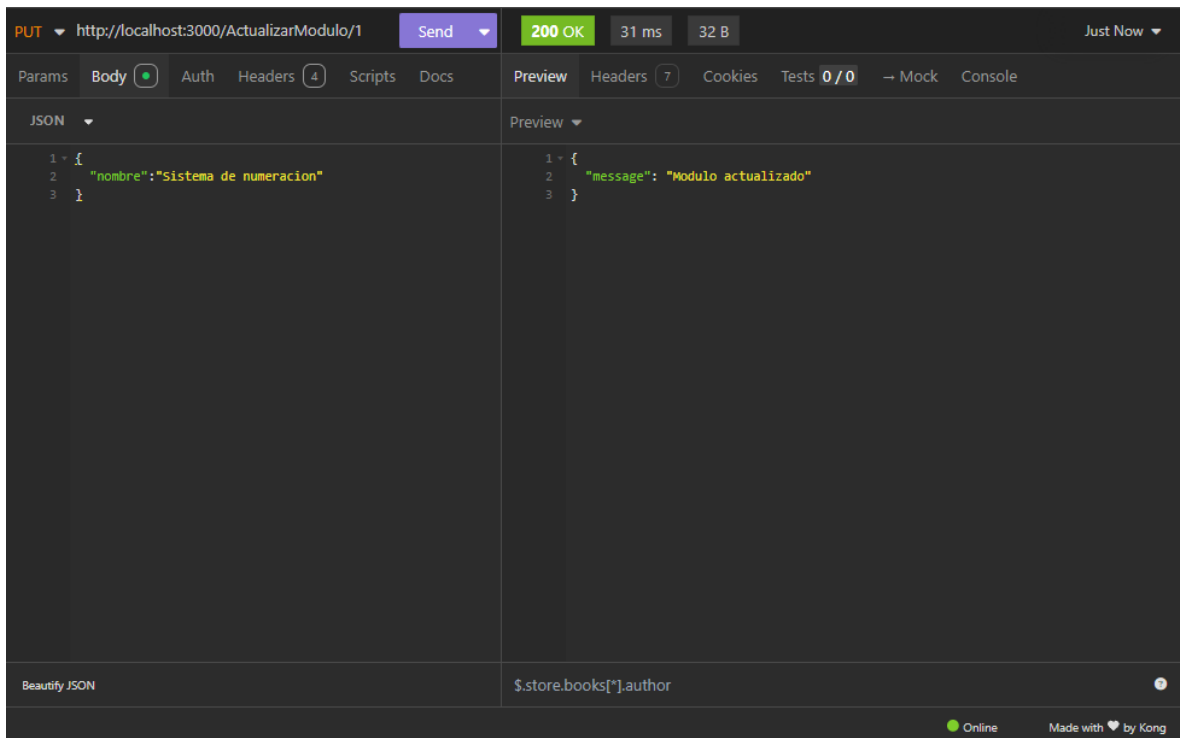
Select a body type from above to send data in the body of a request

Introduction to Insomnia [↗](#)

\$.store.books[0].author

Online Made with ♥ by Kong

Actualizamos el modulo.



PUT ▼ http://localhost:3000/ActualizarModulo/1 Send 200 OK 31 ms 32 B Just Now ▼

Params **Body** ● Auth Headers 4 Scripts Docs **Preview** Headers 7 Cookies Tests 0/0 → Mock Console

JSON ▼

```
1 {  
2   "nombre": "Sistema de numeracion"  
3 }
```

Preview ▼

```
1 {  
2   "message": "Modulo actualizado"  
3 }
```

Beautify JSON

\$.store.books[0].author

Online Made with ♥ by Kong

(En este caso modificamos solo el nombre del módulo)

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos el cambio de nombre del módulo.

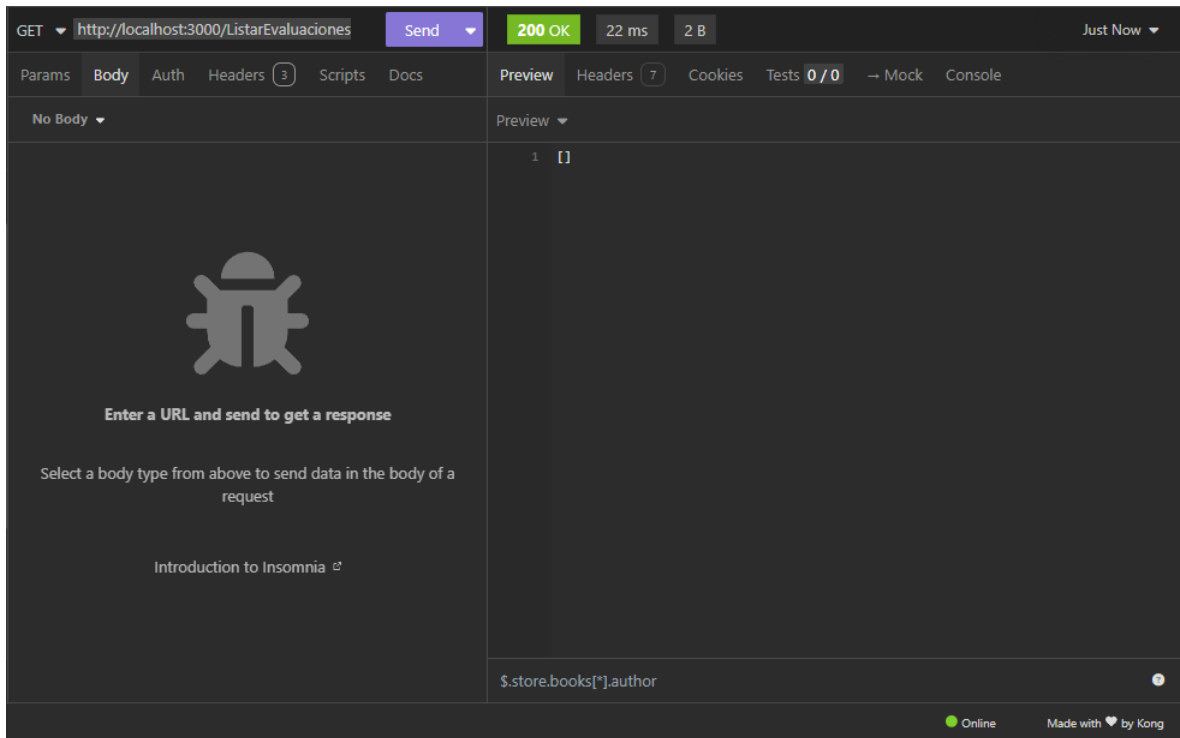
The screenshot displays the Insomnia REST client interface. The top bar shows the request method as GET, the URL as `http://localhost:3000/BuscarModulo/1`, and the status as 200 OK. The response body is a JSON object with the following structure:

```
1 {  
2   "idModulo": 1,  
3   "nombre": "Sistema de numeracion",  
4   "duracion": "3 semanas",  
5   "contenido": "Introduccion.",  
6   "cantidadEvaluaciones": 0,  
7   "estado": "ACTIVO",  
8   "idcurso": 1  
9 }
```

The left sidebar contains a search icon and the text "Enter a URL and send to get a response". The bottom right corner indicates the application is online and was made with Kong.

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.
-Evaluaciones

Listamos las evaluaciones



GET <http://localhost:3000/ListarEvaluaciones> Send

200 OK 22 ms 2 B Just Now

Params Body Auth Headers 3 Scripts Docs Preview Headers 7 Cookies Tests 0/0 → Mock Console

No Body

Preview

```
1 []
```

Enter a URL and send to get a response

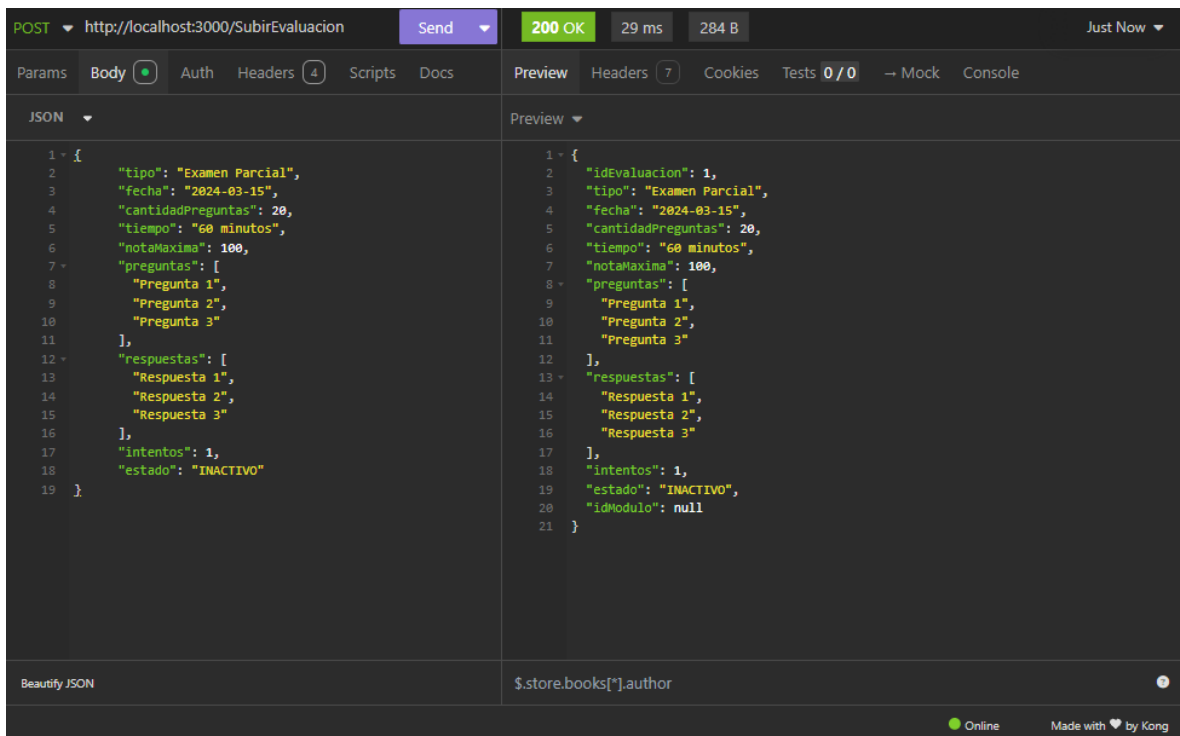
Select a body type from above to send data in the body of a request

Introduction to Insomnia

\$.store.books[*].author

Online Made with by Kong

Creamos una nueva evaluación



POST <http://localhost:3000/SubirEvaluacion> Send

200 OK 29 ms 284 B Just Now

Params Body Auth Headers 4 Scripts Docs Preview Headers 7 Cookies Tests 0/0 → Mock Console

JSON

```
1 {
2   "tipo": "Examen Parcial",
3   "fecha": "2024-03-15",
4   "cantidadPreguntas": 20,
5   "tiempo": "60 minutos",
6   "notaMaxima": 100,
7   "preguntas": [
8     "Pregunta 1",
9     "Pregunta 2",
10    "Pregunta 3"
11  ],
12  "respuestas": [
13    "Respuesta 1",
14    "Respuesta 2",
15    "Respuesta 3"
16  ],
17  "intentos": 1,
18  "estado": "INACTIVO"
19 }
```

Preview

```
1 {
2   "idEvaluacion": 1,
3   "tipo": "Examen Parcial",
4   "fecha": "2024-03-15",
5   "cantidadPreguntas": 20,
6   "tiempo": "60 minutos",
7   "notaMaxima": 100,
8   "preguntas": [
9     "Pregunta 1",
10    "Pregunta 2",
11    "Pregunta 3"
12  ],
13  "respuestas": [
14    "Respuesta 1",
15    "Respuesta 2",
16    "Respuesta 3"
17  ],
18  "intentos": 1,
19  "estado": "INACTIVO",
20  "idModulo": null
21 }
```

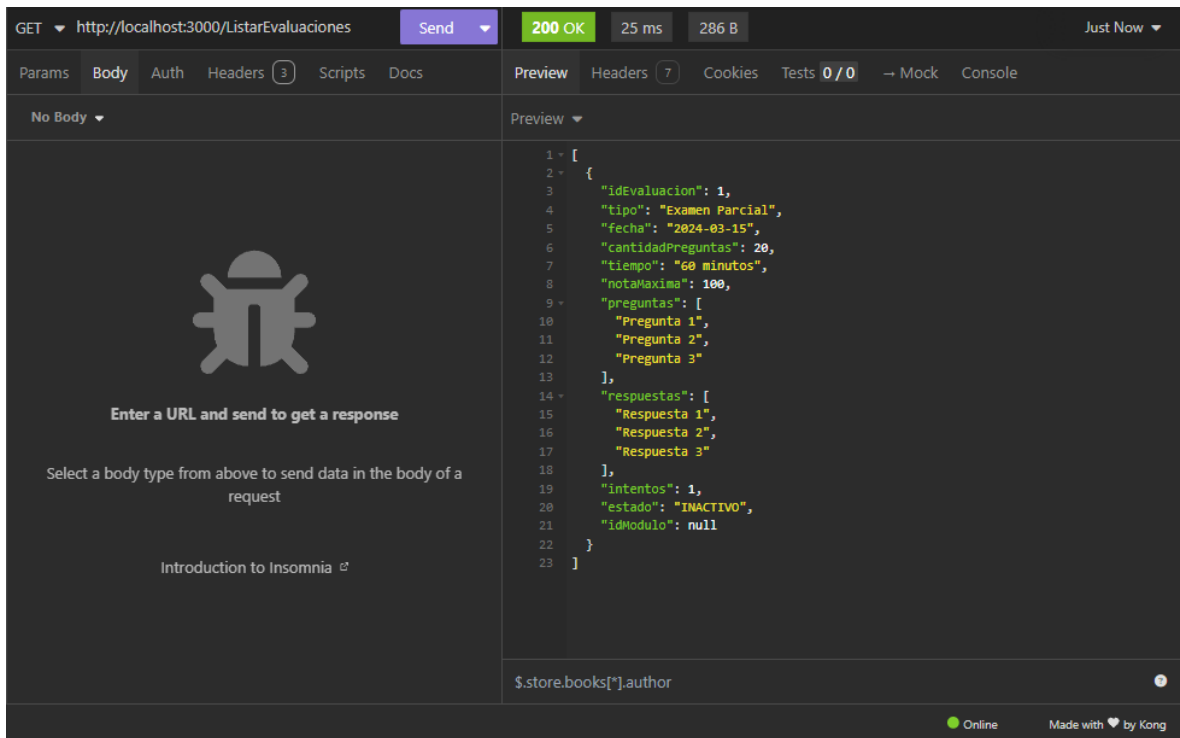
Beautify JSON

\$.store.books[*].author

Online Made with by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos la creación de la evaluación



GET ▼ http://localhost:3000/ListarEvaluaciones Send 200 OK 25 ms 286 B Just Now ▼

Params **Body** Auth Headers 3 Scripts Docs

Preview Headers 7 Cookies Tests 0/0 → Mock Console

No Body ▼

Preview ▼

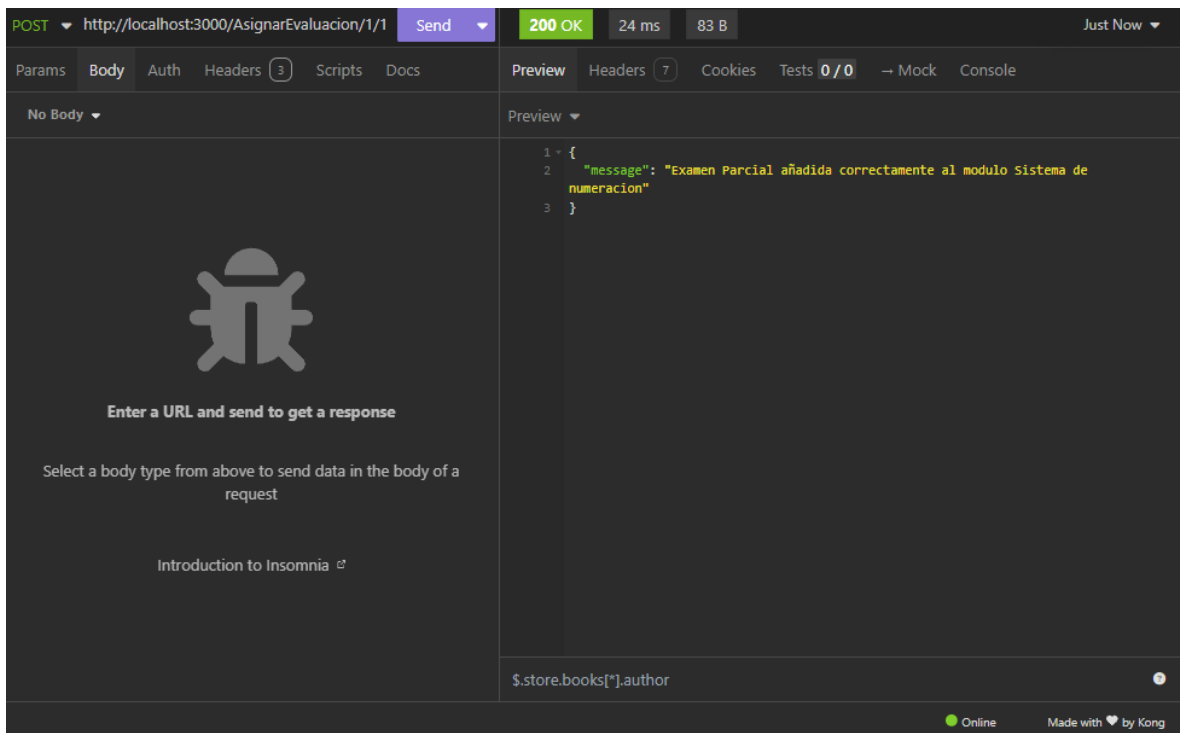
```
1 - [  
2 - {  
3 -   "idevaluacion": 1,  
4 -   "tipo": "Examen Parcial",  
5 -   "fecha": "2024-03-15",  
6 -   "cantidadPreguntas": 20,  
7 -   "tiempo": "60 minutos",  
8 -   "notaMaxima": 100,  
9 -   "preguntas": [  
10 -     "Pregunta 1",  
11 -     "Pregunta 2",  
12 -     "Pregunta 3"  
13 -   ],  
14 -   "respuestas": [  
15 -     "Respuesta 1",  
16 -     "Respuesta 2",  
17 -     "Respuesta 3"  
18 -   ],  
19 -   "intentos": 1,  
20 -   "estado": "INACTIVO",  
21 -   "idModulo": null  
22 - }  
23 - ]
```

\$.store.books[*].author

Online Made with ♥ by Kong

(Vemos que la evaluación no está asignada a ningún modulo)

Asignamos la Evaluación a un modulo



POST ▼ http://localhost:3000/AsignarEvaluacion/1/1 Send 200 OK 24 ms 83 B Just Now ▼

Params **Body** Auth Headers 3 Scripts Docs

Preview Headers 7 Cookies Tests 0/0 → Mock Console

No Body ▼

Preview ▼

```
1 - {  
2 -   "message": "Examen Parcial añadida correctamente al modulo sistema de  
   numeracion"  
3 - }
```

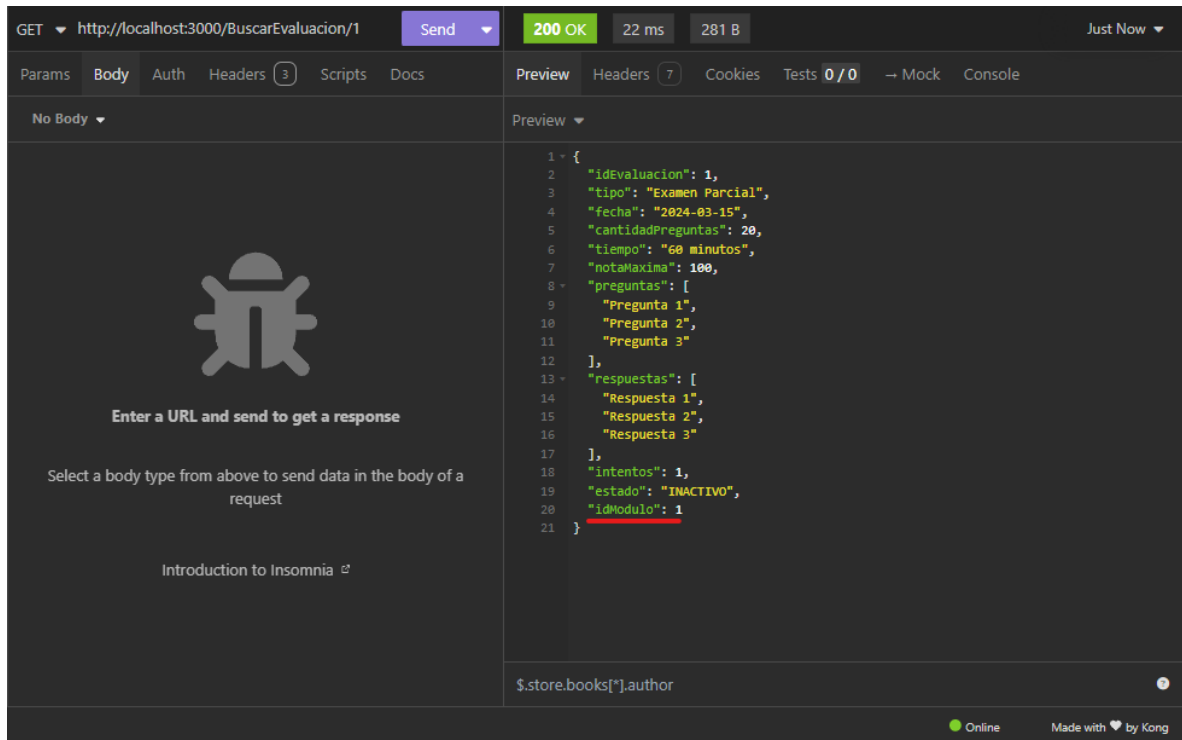
\$.store.books[*].author

Online Made with ♥ by Kong

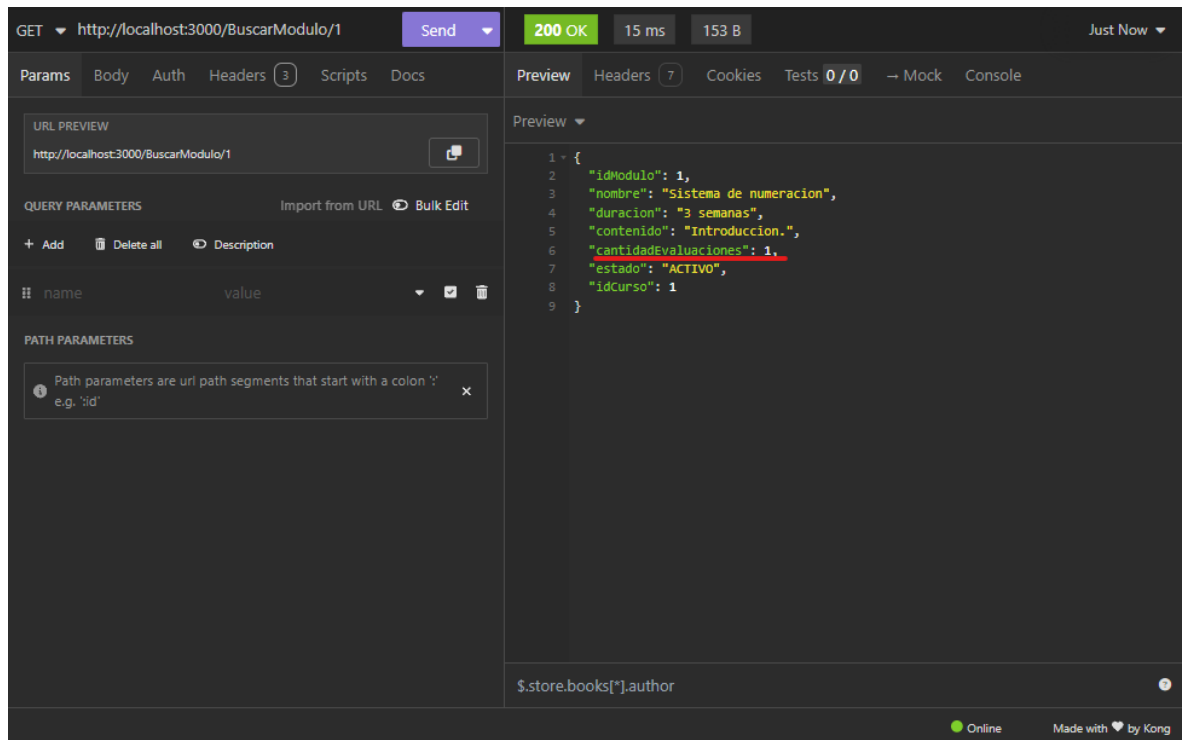
(En este caso asignamos al módulo con id 1 la evaluación con id 1)

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos que la evaluación se añadió correctamente al módulo por su id específico.

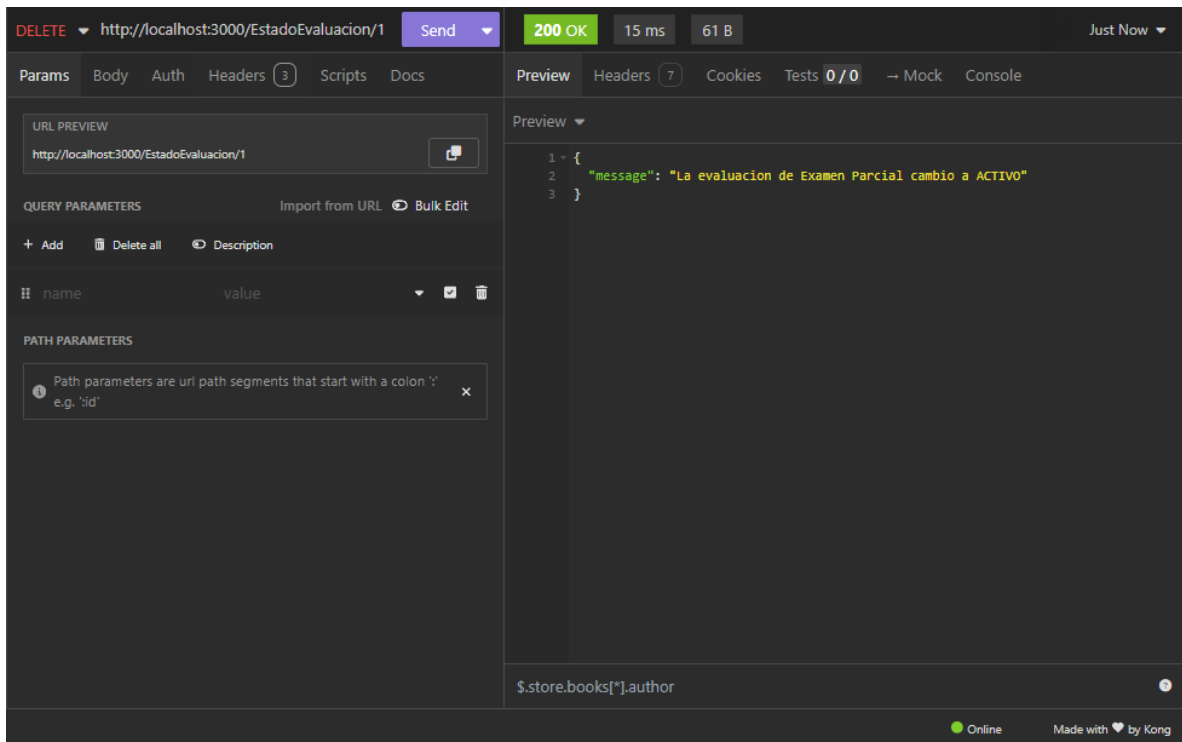


Verificamos la cantidad de evaluaciones del módulo.

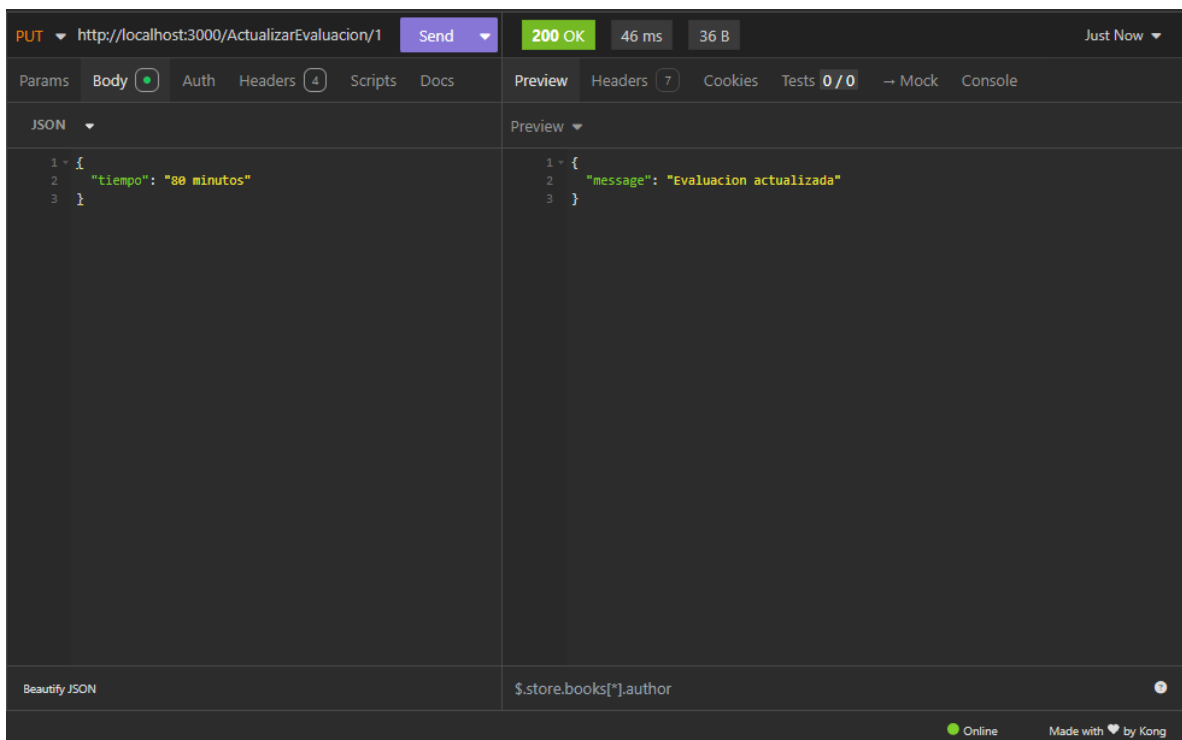


Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Habilitamos la evaluación por el id específico.



Actualizamos la evaluación.



(En este caso solo actualizamos el tiempo de evaluación)

Alumno: Ortega Marcos.

Carrera Técnico Programador.

Curso: 6to 2da.

Verificamos los cambios hechos.

GET ▼ http://localhost:3000/BuscarEvaluacion/1 Send 200 OK 22 ms 279 B Just Now ▼

Params Body Auth Headers 3 Scripts Docs

URL PREVIEW
http://localhost:3000/BuscarEvaluacion/1 Copy

QUERY PARAMETERS Import from URL Bulk Edit

+ Add 🗑 Delete all 📄 Description

name	value
------	-------

PATH PARAMETERS

? Path parameters are url path segments that start with a colon ':'
e.g. ':id' ×

Preview ▼ Headers 7 Cookies Tests 0/0 → Mock Console

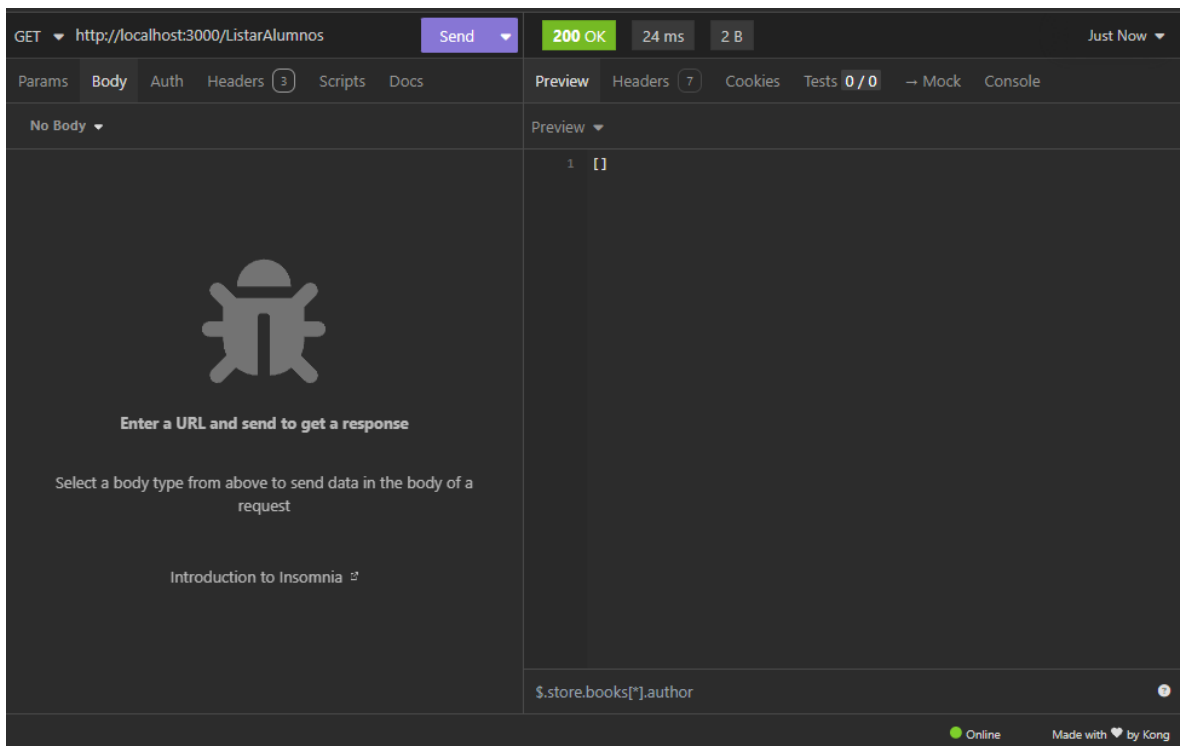
```
1 {  
2   "idEvaluacion": 1,  
3   "tipo": "Examen Parcial",  
4   "fecha": "2024-03-15",  
5   "cantidadPreguntas": 20,  
6   "tiempo": "80 minutos",  
7   "notaMaxima": 100,  
8   "preguntas": [  
9     "Pregunta 1",  
10    "Pregunta 2",  
11    "Pregunta 3"  
12  ],  
13  "respuestas": [  
14    "Respuesta 1",  
15    "Respuesta 2",  
16    "Respuesta 3"  
17  ],  
18  "intentos": 1,  
19  "estado": "ACTIVO",  
20  "IDModulo": 1  
21 }
```

\$.store.books[*].author ?

Online Made with ♥ by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.
-Alumnos

Listar alumnos.



GET ▼ http://localhost:3000/ListarAlumnos Send 200 OK 24 ms 2 B Just Now ▼

Params **Body** Auth Headers 3 Scripts Docs

Preview Headers 7 Cookies Tests 0/0 → Mock Console

No Body ▼

Preview ▼

```
1 []
```

Enter a URL and send to get a response

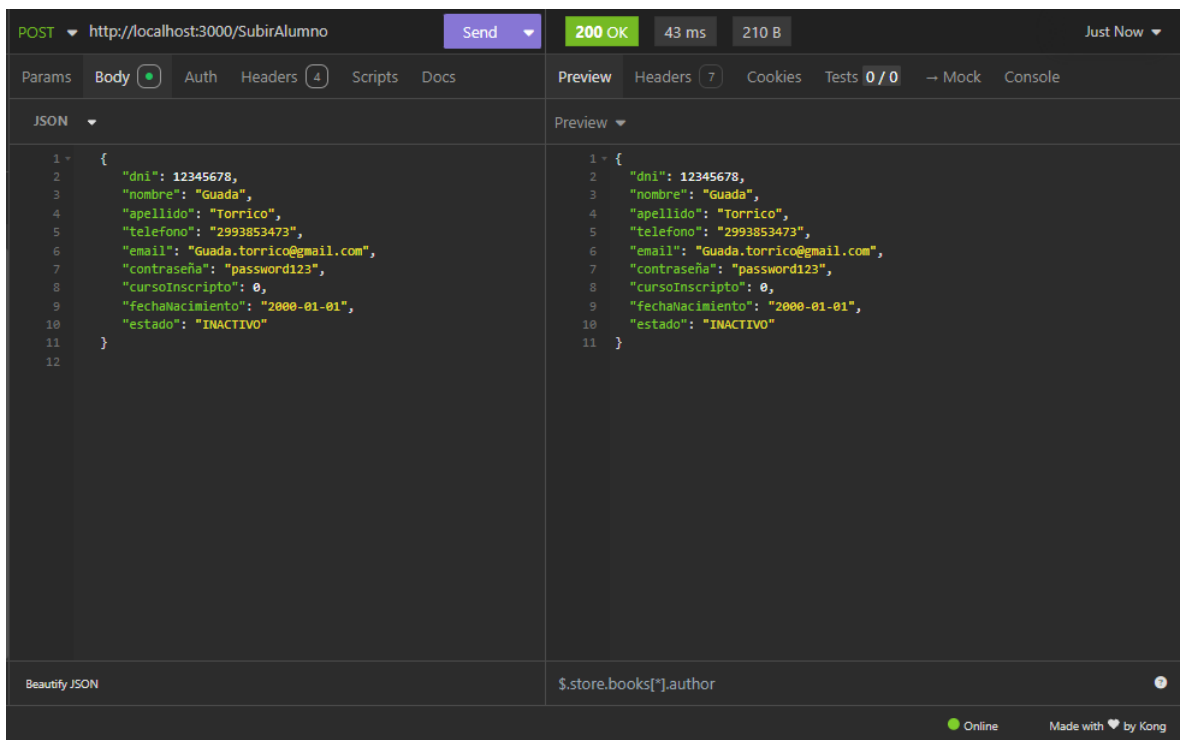
Select a body type from above to send data in the body of a request

Introduction to Insomnia ↗

\$.store.books[*].author

Online Made with ♥ by Kong

Subimos un nuevo alumno



POST ▼ http://localhost:3000/SubirAlumno Send 200 OK 43 ms 210 B Just Now ▼

Params **Body** ● Auth Headers 4 Scripts Docs

Preview Headers 7 Cookies Tests 0/0 → Mock Console

JSON ▼

```
1 {
2   "dni": 12345678,
3   "nombre": "Guada",
4   "apellido": "Torrico",
5   "telefono": "2993853473",
6   "email": "Guada.torrico@gmail.com",
7   "contraseña": "password123",
8   "cursoInscripto": 0,
9   "fechaNacimiento": "2000-01-01",
10  "estado": "INACTIVO"
11 }
12
```

Preview ▼

```
1 {
2   "dni": 12345678,
3   "nombre": "Guada",
4   "apellido": "Torrico",
5   "telefono": "2993853473",
6   "email": "Guada.torrico@gmail.com",
7   "contraseña": "password123",
8   "cursoInscripto": 0,
9   "fechaNacimiento": "2000-01-01",
10  "estado": "INACTIVO"
11 }
```

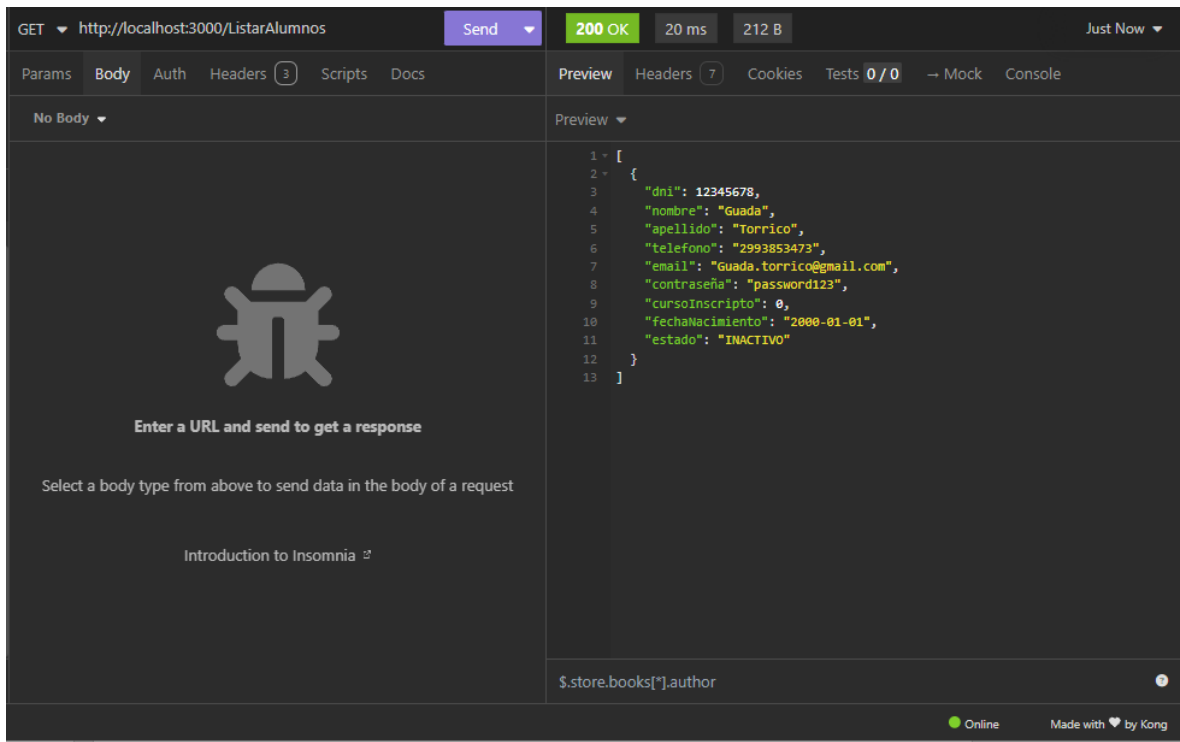
Beautify JSON

\$.store.books[*].author

Online Made with ♥ by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos que el alumno si se subió correctamente.



GET http://localhost:3000/ListarAlumnos Send 200 OK 20 ms 212 B Just Now

Params Body Auth Headers 3 Scripts Docs Preview Headers 7 Cookies Tests 0/0 → Mock Console

No Body

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

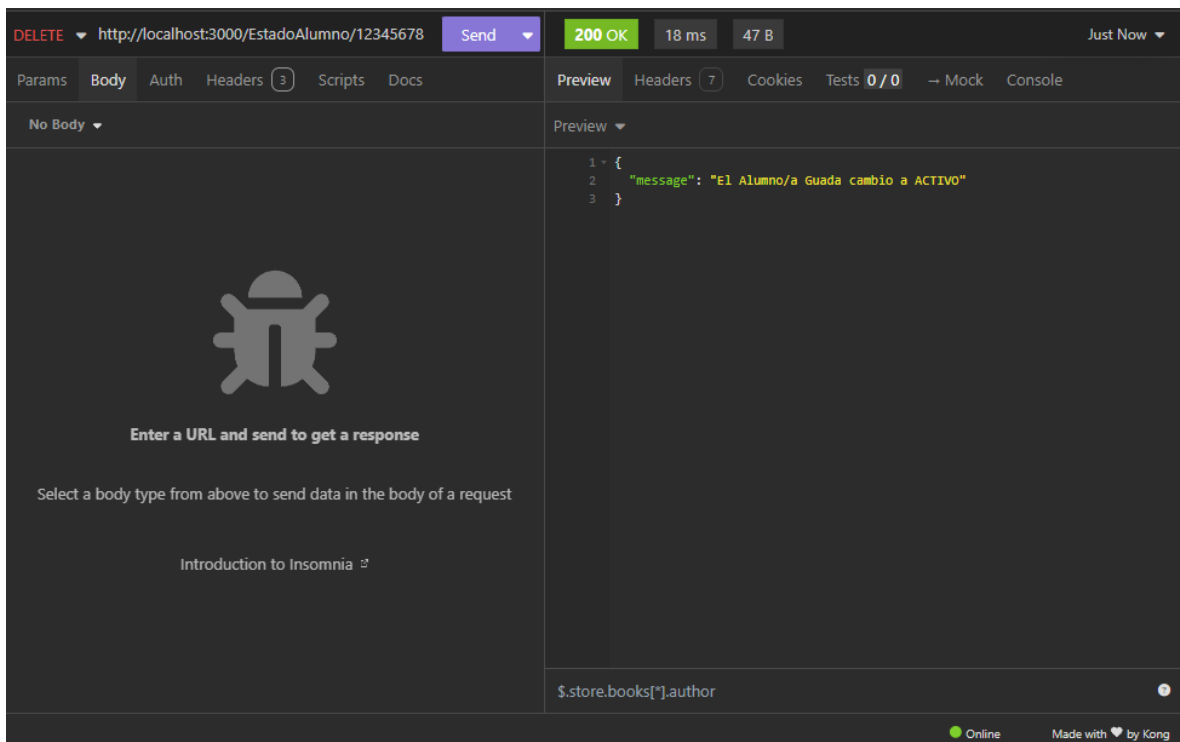
Introduction to Insomnia

```
1 [
2   {
3     "dni": 12345678,
4     "nombre": "Guada",
5     "apellido": "Torrico",
6     "telefono": "2993853473",
7     "email": "Guada.torrico@gmail.com",
8     "contraseña": "password123",
9     "cursoInscrito": 0,
10    "fechaNacimiento": "2000-01-01",
11    "estado": "INACTIVO"
12  }
13 ]
```

\$.store.books[*].author

Online Made with ♥ by Kong

Habilitamos el alumno por su DNI.



DELETE http://localhost:3000/EstadoAlumno/12345678 Send 200 OK 18 ms 47 B Just Now

Params Body Auth Headers 3 Scripts Docs Preview Headers 7 Cookies Tests 0/0 → Mock Console

No Body

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Introduction to Insomnia

```
1 {
2   "message": "El Alumno/a Guada cambio a ACTIVO"
3 }
```

\$.store.books[*].author

Online Made with ♥ by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Actualizamos el alumno por su DNI.

The screenshot shows the Insomnia client interface for a PUT request. The URL is `http://localhost:3000/ActualizarAlumno/12345678`. The request body is a JSON object: `{ "nombre": "Guadalupe" }`. The response is a 200 OK status with a response time of 22 ms and a body size of 34 B. The response body is a JSON object: `{ "message": "Alumno/a Actualizado" }`.

```
PUT http://localhost:3000/ActualizarAlumno/12345678
{
  "nombre": "Guadalupe"
}
```

```
200 OK
{
  "message": "Alumno/a Actualizado"
}
```

(En este caso se cambió solo el nombre del alumno)

Verificamos los cambios realizados por DNI del alumno.

The screenshot shows the Insomnia client interface for a GET request. The URL is `http://localhost:3000/BuscarAlumno/12345678`. The response is a 200 OK status with a response time of 15 ms and a body size of 212 B. The response body is a JSON object containing student details, with several fields highlighted in red: `"nombre": "Guadalupe", "apellido": "Torrico", "telefono": "2993853473", "email": "Guada.torrico@gmail.com", "contraseña": "password123", "cursoInscripto": 0, "fechaNacimiento": "2000-01-01", "estado": "ACTIVO"`.

```
GET http://localhost:3000/BuscarAlumno/12345678
```

```
200 OK
{
  "dni": 12345678,
  "nombre": "Guadalupe",
  "apellido": "Torrico",
  "telefono": "2993853473",
  "email": "Guada.torrico@gmail.com",
  "contraseña": "password123",
  "cursoInscripto": 0,
  "fechaNacimiento": "2000-01-01",
  "estado": "ACTIVO"
}
```

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

(El alumno no está inscripto a ningún curso)

Inscribir alumno a un curso.

The screenshot shows the Insomnia client interface. The top bar indicates a POST request to `http://localhost:3000/InscribirAlumno/12345678/1` with a status of 200 OK, 28 ms response time, and 174 B body size. The left sidebar shows the URL, query parameters, and path parameters. The main area displays the JSON response:

```
1 {
2   "message": "Alumno inscrito correctamente:",
3   "inscripcion": {
4     "alumnoDni": 12345678,
5     "idCurso": 1,
6     "estadoInscripcion": "Completa",
7     "fechaInscripcion": "2024-01-11",
8     "pago": "Gratis"
9   }
10 }
```

(El alumno con el DNI 12345678 se inscribió al curso con id 1)

Verificamos los cambios en el alumno y en el curso.

The screenshot shows the Insomnia client interface. The top bar indicates a GET request to `http://localhost:3000/BuscarAlumno/12345678` with a status of 200 OK, 16 ms response time, and 212 B body size. The left sidebar shows the URL, query parameters, and path parameters. The main area displays the JSON response:

```
1 {
2   "dni": 12345678,
3   "nombre": "Guadalupe",
4   "apellido": "Torrico",
5   "telefono": "2993853473",
6   "email": "Guada.torrico@gmail.com",
7   "contraseña": "password123",
8   "cursoInscripto": 1,
9   "fechaNacimiento": "2000-01-01",
10  "estado": "ACTIVO"
11 }
```

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

GET ▼ http://localhost:3000/BuscarCurso/1 Send 200 OK 17 ms 254 B Just Now ▼

ParamsBodyAuthHeaders3ScriptsDocs

URL PREVIEW
http://localhost:3000/BuscarCurso/1 📄

QUERY PARAMETERS
Import from URL 👁 Bulk Edit

+ Add

Delete all

Description

name

value

▼

📄

🗑

PATH PARAMETERS

❗

Path parameters are url path segments that start with a colon ':', e.g. 'id'

×

PreviewHeaders7CookiesTests0/0→MockConsole

Preview ▼

```
1 {  
2   "idCurso": 1,  
3   "nombre": "Organizacion y Arquitectura",  
4   "descripcion": "Curso completamente espumante",  
5   "fechaInicio": "2024-01-10",  
6   "fechaFin": "2024-06-10",  
7   "duracion": "6 meses",  
8   "cantidadTotalAlumnos": 35,  
9   "estado": "ACTIVO",  
10  "cantidadAlumnos": 1,  
11  "cantidadModulos": 1  
12 }
```

\$.store.books[*].author 🔍

Online

Made with ♥ by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.
-Profesores:

Listamos los profesores

GET ▼ http://localhost:3000/ListarProfesores Send ▼ 200 OK 18 ms 2 B Just Now ▼

Params Body Auth Headers 3 Scripts Docs

URL PREVIEW
http://localhost:3000/ListarProfesores

QUERY PARAMETERS
Import from URL 🔗 Bulk Edit

+ Add 🗑️ Delete all 📄 Description

#	name	value	▼	📄	🗑️
---	------	-------	---	---	----

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview ▼

```
1 []
```

\$store.books[*].author

Online Made with ♥ by Kong

Subir un nuevo Profesor

POST ▼ http://localhost:3000/SubirProfesor Send ▼ 200 OK 29 ms 211 B Just Now ▼

Params Body Auth Headers 4 Scripts Docs

JSON ▼

```
1 {  
2   "dni": "11223354",  
3   "nombre": "Marcos",  
4   "apellido": "Ortega",  
5   "telefono": "2995822312",  
6   "email": "marquitos.ortg@gmail.com",  
7   "contraseña": "profesor123",  
8   "especializacion": "Matematicas",  
9   "cursoAsignado": 1,  
10  "estado": "INACTIVO"  
11 }
```

Preview ▼

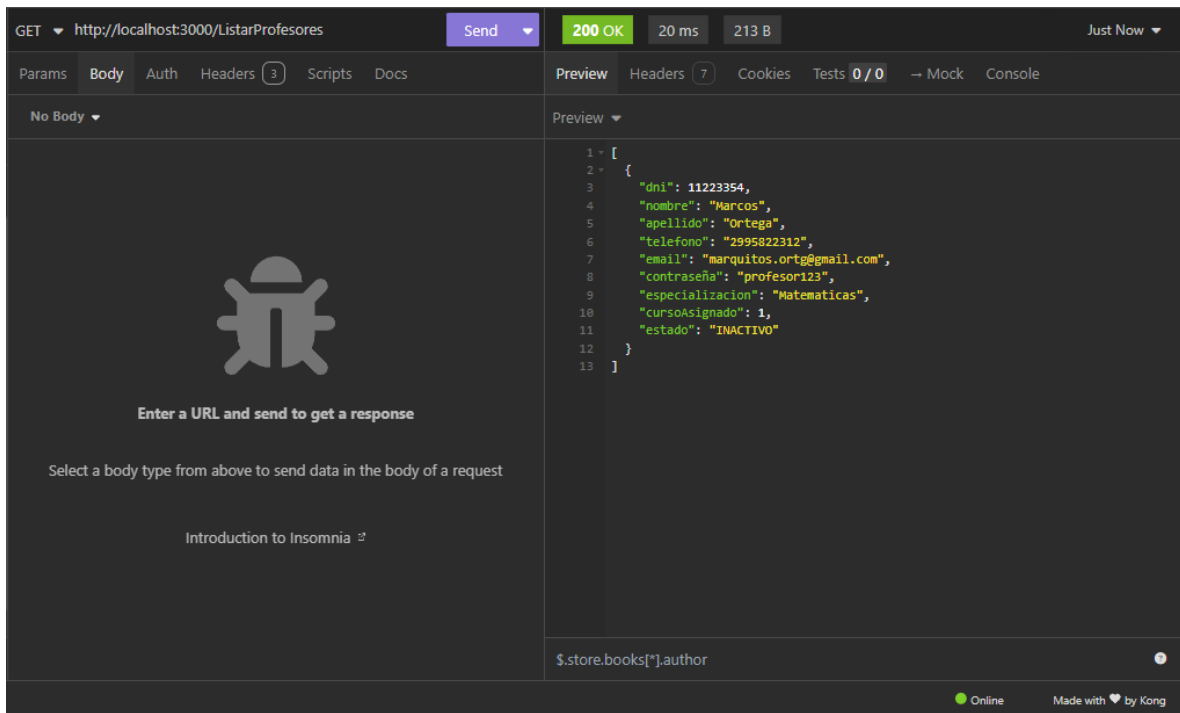
```
1 {  
2   "dni": "11223354",  
3   "nombre": "Marcos",  
4   "apellido": "Ortega",  
5   "telefono": "2995822312",  
6   "email": "marquitos.ortg@gmail.com",  
7   "contraseña": "profesor123",  
8   "especializacion": "Matematicas",  
9   "cursoAsignado": 1,  
10  "estado": "INACTIVO"  
11 }
```

Beautify JSON \$store.books[*].author

Online Made with ♥ by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos la creación del profesor



GET ▼ http://localhost:3000/ListarProfesores Send 200 OK 20 ms 213 B Just Now ▼

Params **Body** Auth Headers 3 Scripts Docs

Preview Headers 7 Cookies Tests 0/0 → Mock Console

No Body ▼

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Introduction to Insomnia [↗](#)

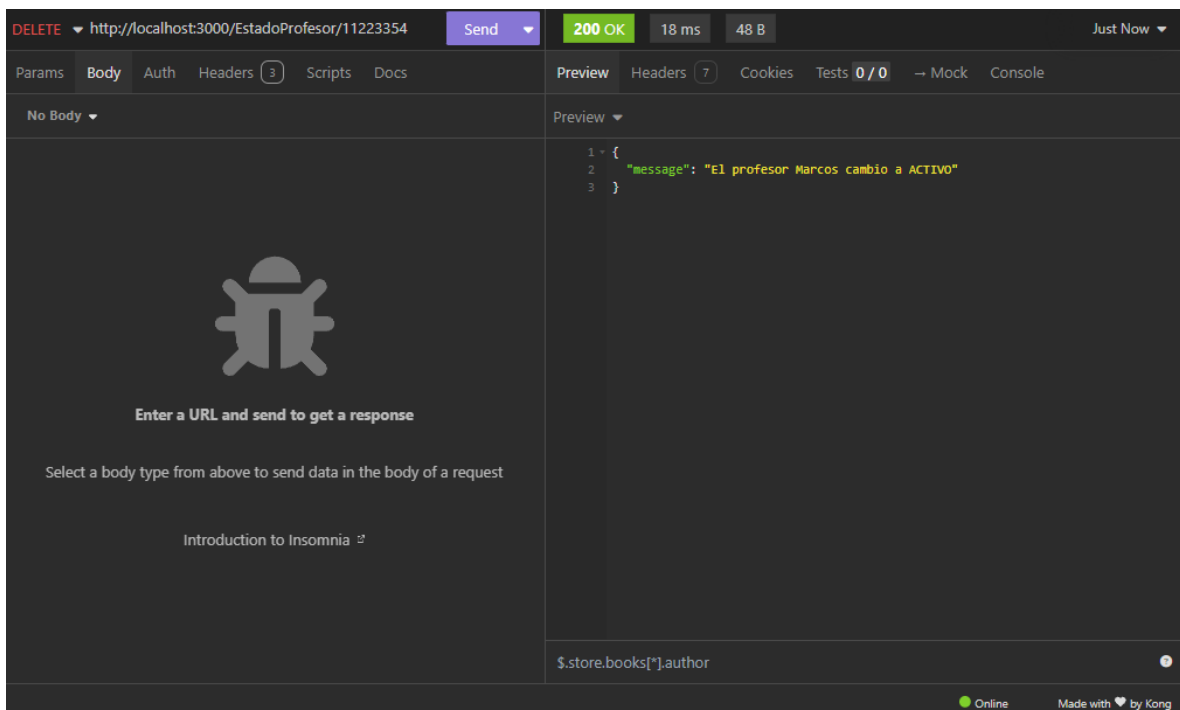
Preview ▼

```
1 {  
2 {  
3   "dni": "11223354",  
4   "nombre": "Marcos",  
5   "apellido": "Ortega",  
6   "telefono": "2995822312",  
7   "email": "marquitos.ortg@gmail.com",  
8   "contraseña": "profesor123",  
9   "especializacion": "Matematicas",  
10  "cursoAsignado": 1,  
11  "estado": "INACTIVO"  
12 }  
13 }
```

\$.store.books[*].author

Online Made with [♥](#) by Kong

Habilitamos el profesor por su DNI.



DELETE ▼ http://localhost:3000/EstadoProfesor/11223354 Send 200 OK 18 ms 48 B Just Now ▼

Params **Body** Auth Headers 3 Scripts Docs

Preview Headers 7 Cookies Tests 0/0 → Mock Console

No Body ▼

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Introduction to Insomnia [↗](#)

Preview ▼

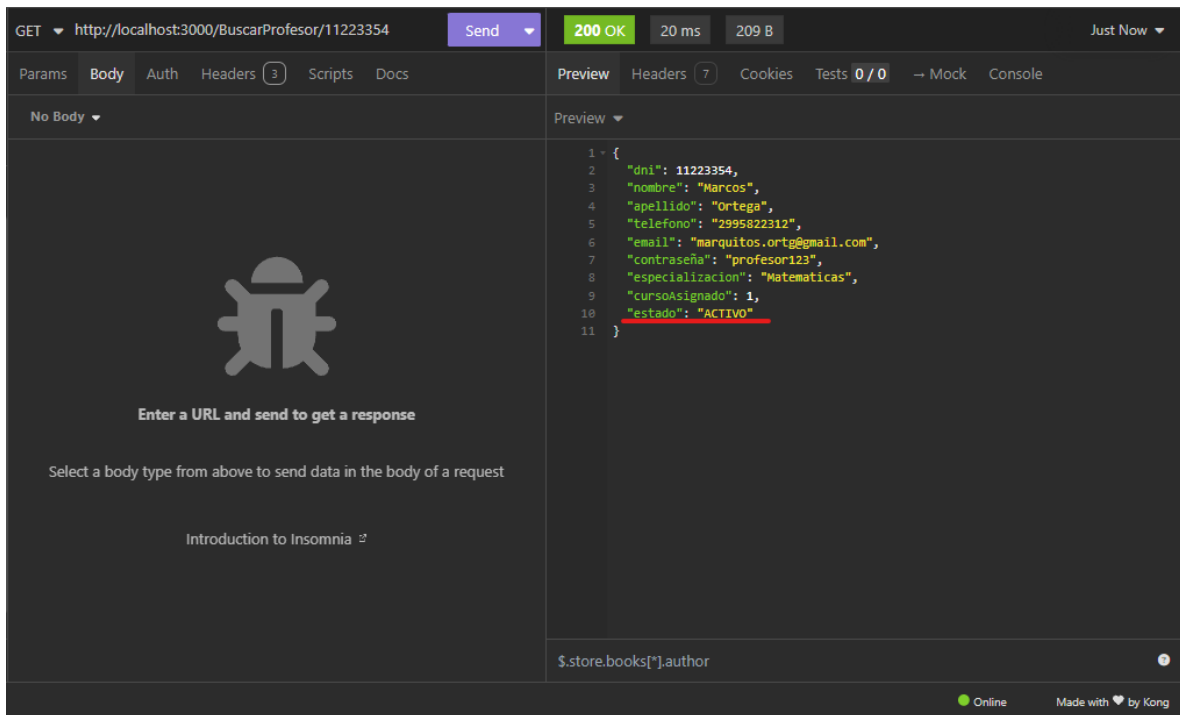
```
1 {  
2   "message": "El profesor Marcos cambio a ACTIVO"  
3 }
```

\$.store.books[*].author

Online Made with [♥](#) by Kong

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos el cambio de estado del profesor por su DNI.



GET ▼ http://localhost:3000/BuscarProfesor/11223354 Send 200 OK 20 ms 209 B Just Now ▼

Params **Body** Auth Headers 3 Scripts Docs

No Body ▼

Preview ▼

```
1 {  
2   "dni": 11223354,  
3   "nombre": "Marcos",  
4   "apellido": "Ortega",  
5   "telefono": "2995822312",  
6   "email": "marquitos.ortg@gmail.com",  
7   "contraseña": "profesor123",  
8   "especializacion": "Matematicas",  
9   "cursoAsignado": 1,  
10  "estado": "ACTIVO"  
11 }
```

Enter a URL and send to get a response

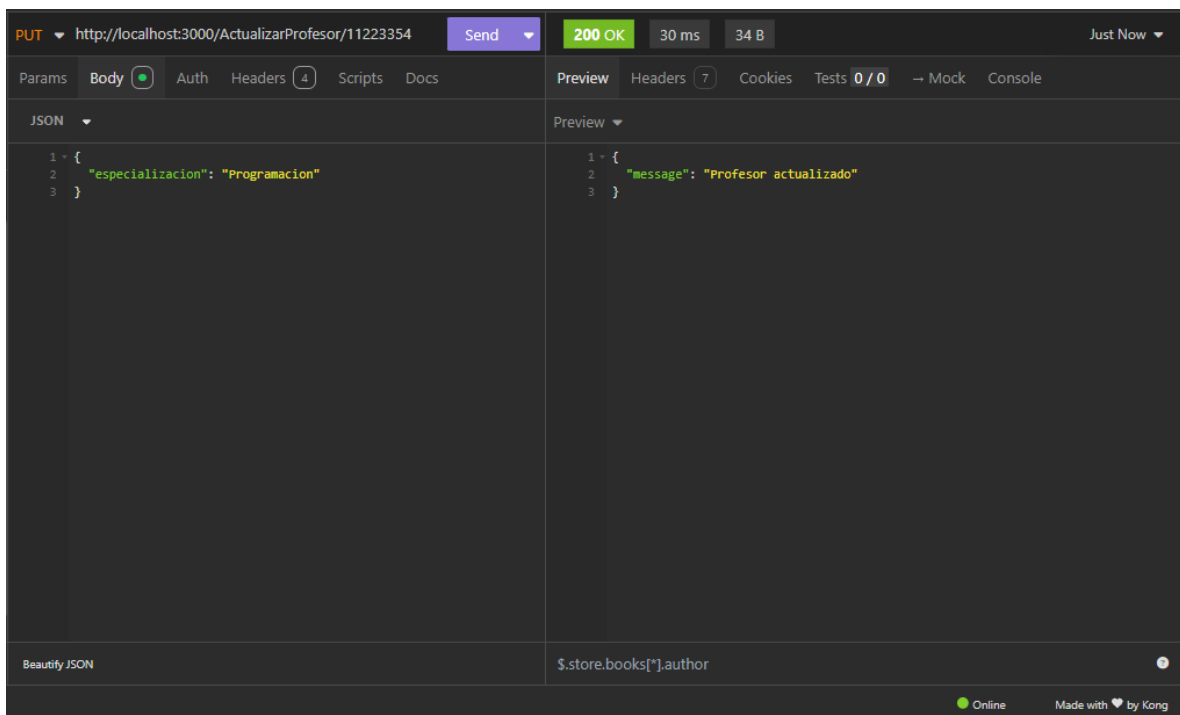
Select a body type from above to send data in the body of a request

Introduction to Insomnia [↗](#)

\$.store.books[*].author

Online Made with ♥ by Kong

Modificamos el profesor por su DNI.



PUT ▼ http://localhost:3000/ActualizarProfesor/11223354 Send 200 OK 30 ms 34 B Just Now ▼

Params **Body** ● Auth Headers 4 Scripts Docs

JSON ▼

```
1 {  
2   "especializacion": "Programacion"  
3 }
```

Preview ▼

```
1 {  
2   "message": "Profesor actualizado"  
3 }
```

Beautify JSON

\$.store.books[*].author

Online Made with ♥ by Kong

(En este caso modificamos su especialización)

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Verificamos si se realizó la modificación.

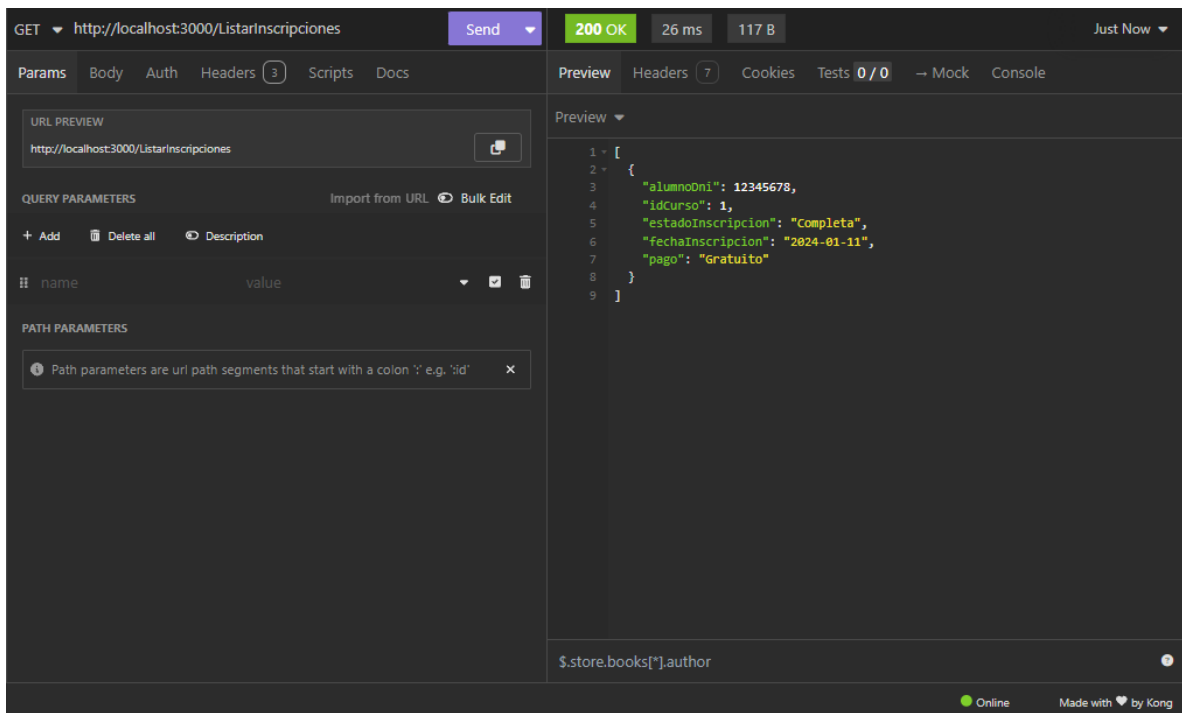
The screenshot shows the Insomnia REST client interface. The top bar displays the method 'GET', the URL 'http://localhost:3000/BuscarProfesor/11223354', and a 'Send' button. The status bar indicates a '200 OK' response with a time of '29 ms' and a size of '210 B'. The left sidebar shows tabs for 'Params', 'Body', 'Auth', 'Headers' (with 3 items), 'Scripts', and 'Docs'. The main area is split into two panes. The left pane, titled 'No Body', contains a large grey bug icon and the text 'Enter a URL and send to get a response', 'Select a body type from above to send data in the body of a request', and a link to 'Introduction to Insomnia'. The right pane, titled 'Preview', shows the JSON response of the request:

```
1 {
2   "dni": 11223354,
3   "nombre": "Marcos",
4   "apellido": "Ortega",
5   "telefono": "2995822312",
6   "email": "marquitos.ortg@gmail.com",
7   "contraseña": "profesor123",
8   "especializacion": "Programacion",
9   "cursoAsignado": 1,
10  "estado": "ACTIVO"
11 }
```

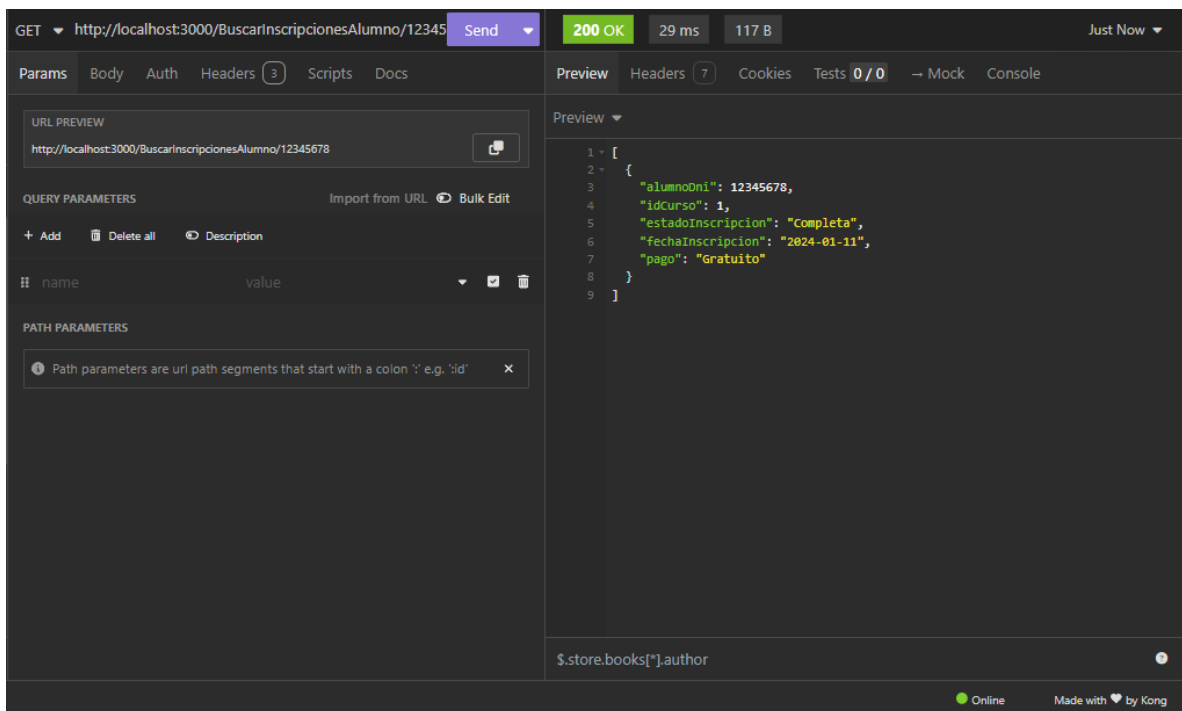
 The value '1' for 'cursoAsignado' is highlighted with a red underline. At the bottom right of the interface, it says 'Online' with a green dot and 'Made with ❤ by Kong'.

Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.
-Inscripciones

Listar las inscripciones



Listar inscripciones por alumno (con su dni)



Alumno: Ortega Marcos.
Carrera Técnico Programador.
Curso: 6to 2da.

Listar inscripciones por curso (con su id)

The screenshot displays the Kong API client interface. The top bar shows the method **GET**, the URL **http://localhost:3000/BuscarInscripcionesCurso/1**, and a **Send** button. To the right, the response status is **200 OK** with a response time of **13 ms** and a size of **117 B**. Below the top bar, the left sidebar contains tabs for **Params**, **Body**, **Auth**, **Headers** (3), **Scripts**, and **Docs**. The **Params** tab is active, showing **QUERY PARAMETERS** and **PATH PARAMETERS**. The **QUERY PARAMETERS** section has a table with one parameter: **name** with **value**. The **PATH PARAMETERS** section has a note: "Path parameters are uri path segments that start with a colon ':' e.g. ':id'". The right sidebar contains tabs for **Preview**, **Headers** (7), **Cookies**, **Tests** (0/0), **Mock**, and **Console**. The **Preview** tab is active, showing the response body as a JSON array with one object. The bottom status bar indicates the client is **Online** and was made with Kong.

GET **http://localhost:3000/BuscarInscripcionesCurso/1** **Send** **200 OK** 13 ms 117 B Just Now

Params Body Auth Headers 3 Scripts Docs

URL PREVIEW
http://localhost:3000/BuscarInscripcionesCurso/1

QUERY PARAMETERS Import from URL Bulk Edit

+ Add Delete all Description

name	value
name	value

PATH PARAMETERS

Path parameters are uri path segments that start with a colon ':' e.g. ':id'

Preview Headers 7 Cookies Tests 0/0 Mock Console

```
1 - [  
2 - {  
3   "alumnoDni": 12345678,  
4   "idCurso": 1,  
5   "estadoInscripcion": "Completa",  
6   "fechaInscripcion": "2024-01-11",  
7   "pago": "Gratis"  
8 }  
9 ]
```

\$.store.books[*].author

Online Made with Kong