Artículo
Muhammad Waseem · Dic 22, 2021 Lectura de 3 min

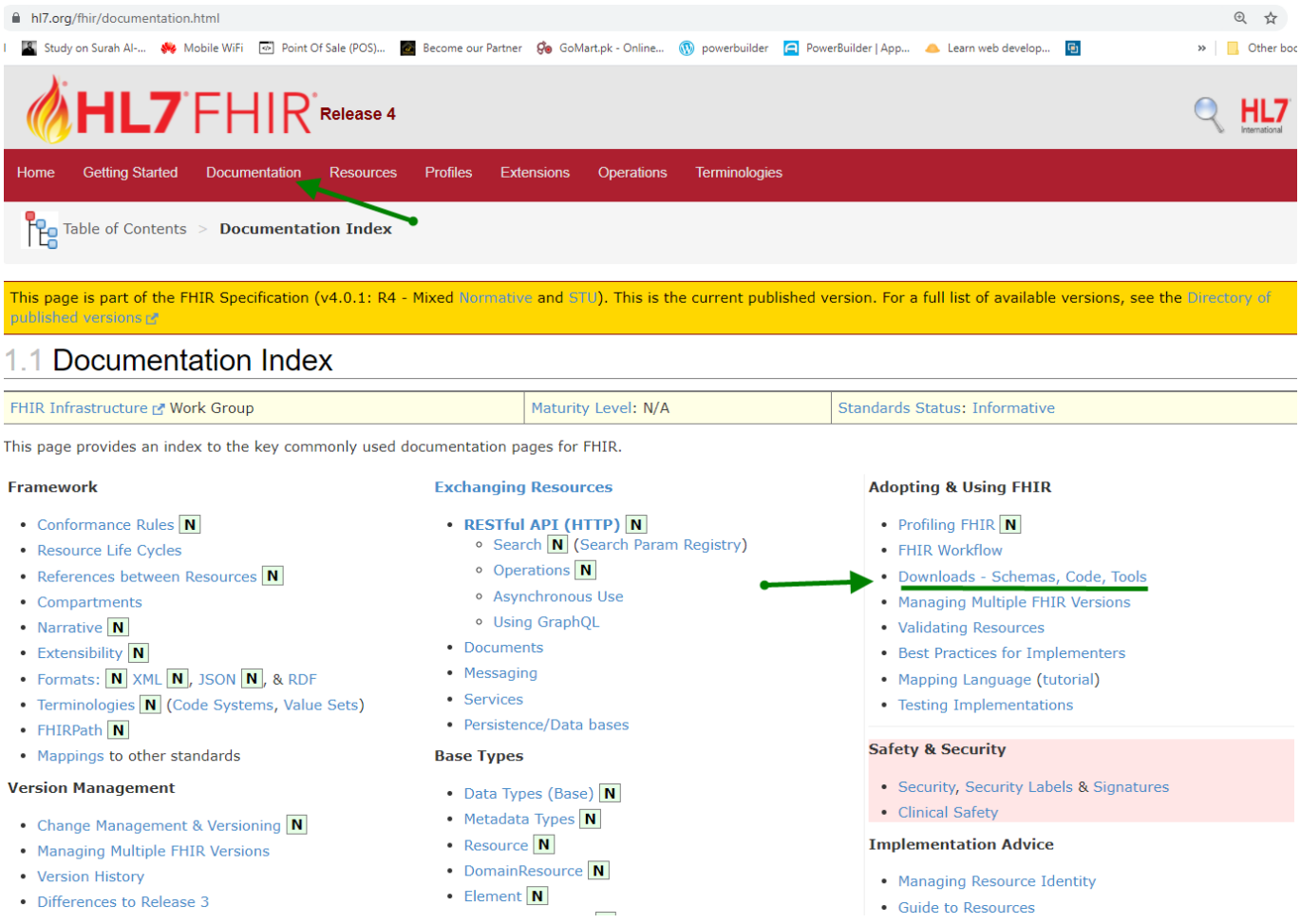
Cómo crear y validar cualquier recurso FHIR usando el esquema FHIR con la ayuda de IntelliSense y la función de autocompletado en VS Code

La interoperabilidad en la asistencia sanitaria es esencial para mejorar la atención a los pacientes, reducir los costes de los proveedores de atención médica y ofrecer una imagen más precisa a los proveedores. Pero con tantos sistemas diferentes, los datos se presentan en diferentes formatos. Se han creado muchos estándares para tratar de resolver este problema, incluyendo HL7v2, HL7v3 y CDA, pero cada una tiene sus limitaciones.

FHIR, o Fast Healthcare Interoperability Resources, es un estándar para el intercambio de datos de salud, que tiene como objetivo resolver estos problemas. Fue desarrollado por Health Level Seven International (HL7), una organización que también desarrolló HL7v2, HL7v3 y CDA.

En este artículo descubriremos cómo crear y validar recursos FHIR usando el esquema FHIR con la ayuda de IntelliSense y la función de autocompletado en VS Code.

Paso 1: Descargar el archivo JSON Schema para Validación de Recursos desde la web oficial de FHIR: https://www.hl7.org/fhir/



hl7.org/fhir/downloads.html

Study on Surah Al-... Mobile WiFi Point Of Sale (POS)... Become our Partner GoMart.pk - Online... powerbuilder PowerBuilder | App... Learn web develop... Other

This page is part of the FHIR Specification (v4.0.1: R4 - Mixed Normative and STU). This is the current published version. For a full list of available versions, see the Directory of published versions

7.1 Downloads

FHIR Infrastructure Work Group Maturity Level: N/A Standards Status: Informative

Specification Downloads

FHIR Definitions All the value sets, profiles, etc. defined as part of the FHIR specification, and the included implementation guides:

- XML
- JSON
- Forge: Special version of definitions for Forge (temporary)

This is the master set of definitions that should be the first choice whenever generating any implementation artifacts. All the other forms below include only subsets of the information available in these definition files, and do not contain all of the rules about what makes resources valid. Implementers will still need to be familiar with the content of the specification and with any profiles that apply to the resources in order to make a conformant implementation.

XML

- Examples - all the example resources in XML format
- Validation Schemas (includes support schemas, resource schemas, modular & combined schemas, and Schematrons)
- Code Generation Schemas (see notes about code-generation schemas)
  Note that names relevant for code generation, including resource names, element & slice names, codes, etc. may collide with reserved words in the relevant target language, and code generators will need to handle this

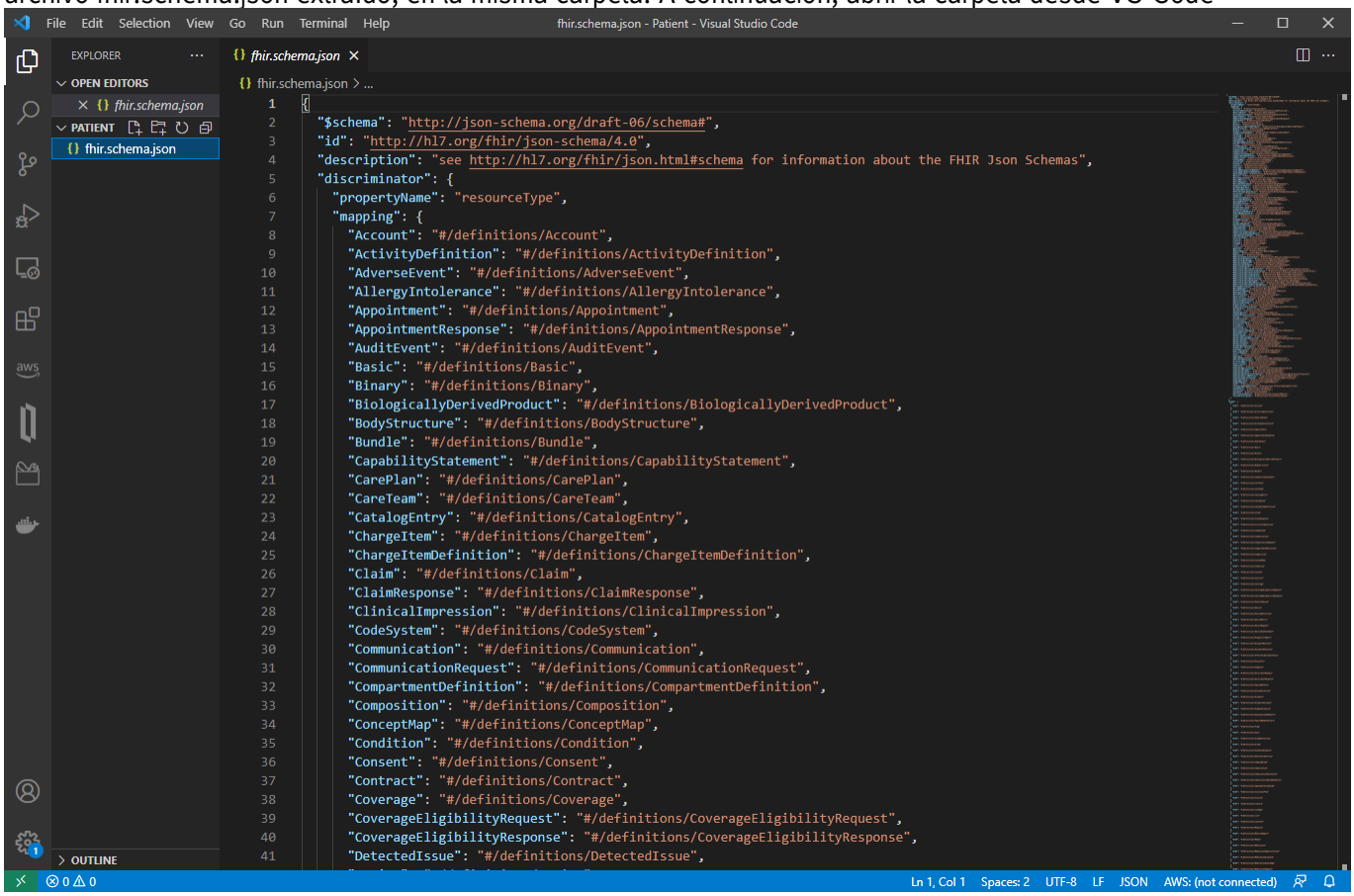
JSON

- Examples - all the example resources in JSON format
- JSON Schema (Needs JSON Schema draft-06 or more recent)
- Examples - all the example resources in ND-JSON format (bulk data format)

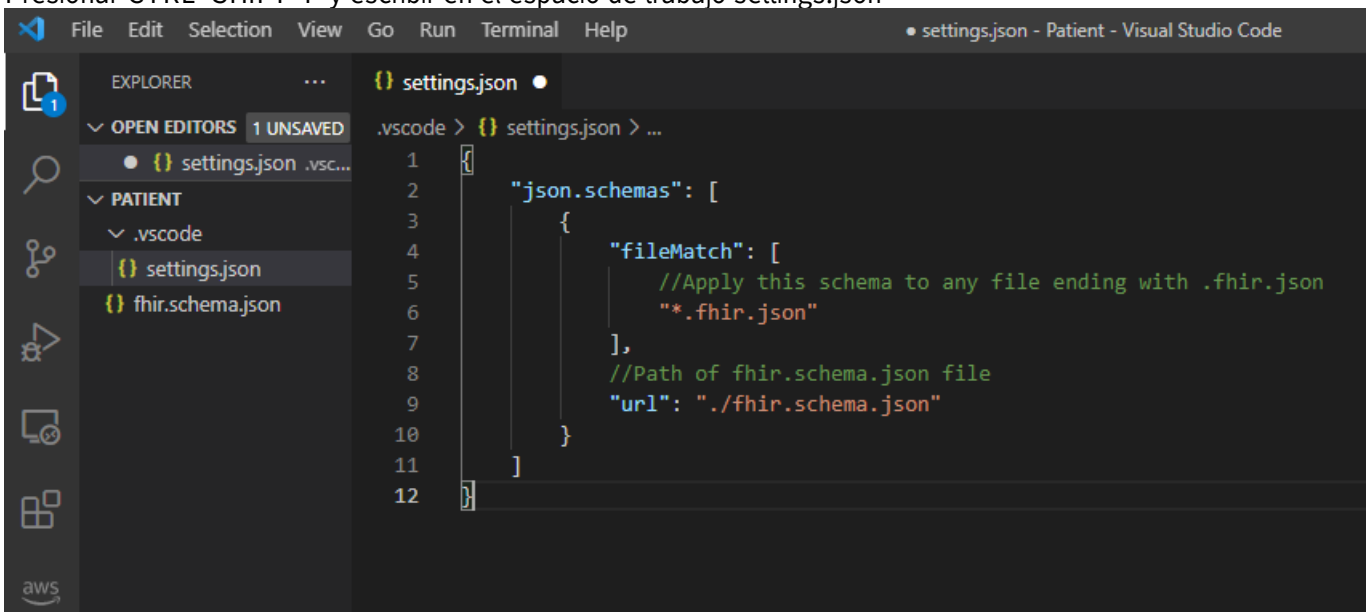
RDF

- Turtle Examples - all the example resources in Turtle format
- ShEx Schemas - ShEx definitions for validating RDF resources
- Definitions - the formal definitions that define the predicates and classes used in the RDF format (not up to date)

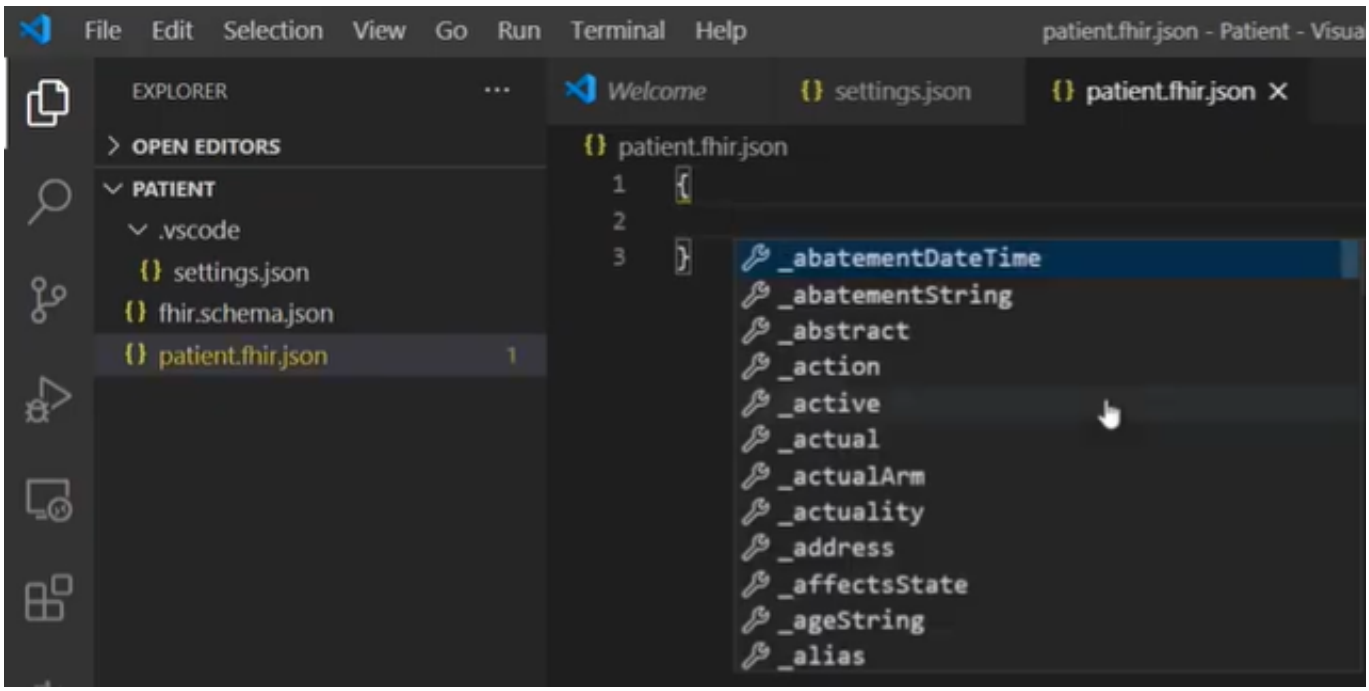
Paso 2: Crear una carpeta (en este ejemplo estoy usando la carpeta Patient y el recurso Patient) y copiar el archivo fhir.schema.json extraído, en la misma carpeta. A continuación, abrir la carpeta desde VS Code



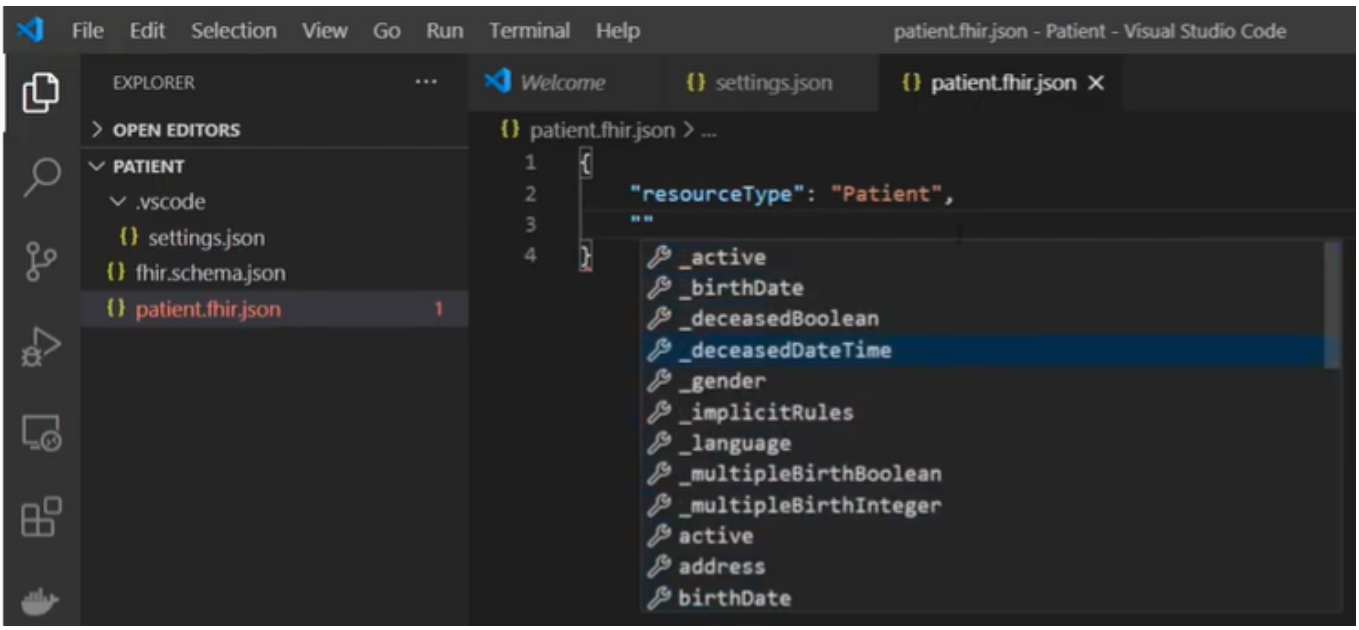
Paso 3: Configurar el código VS para reconocer el esquema FHIR modificando el archivo setting.json.
Presionar CTRL+SHIFT+P y escribir en el espacio de trabajo settings.json



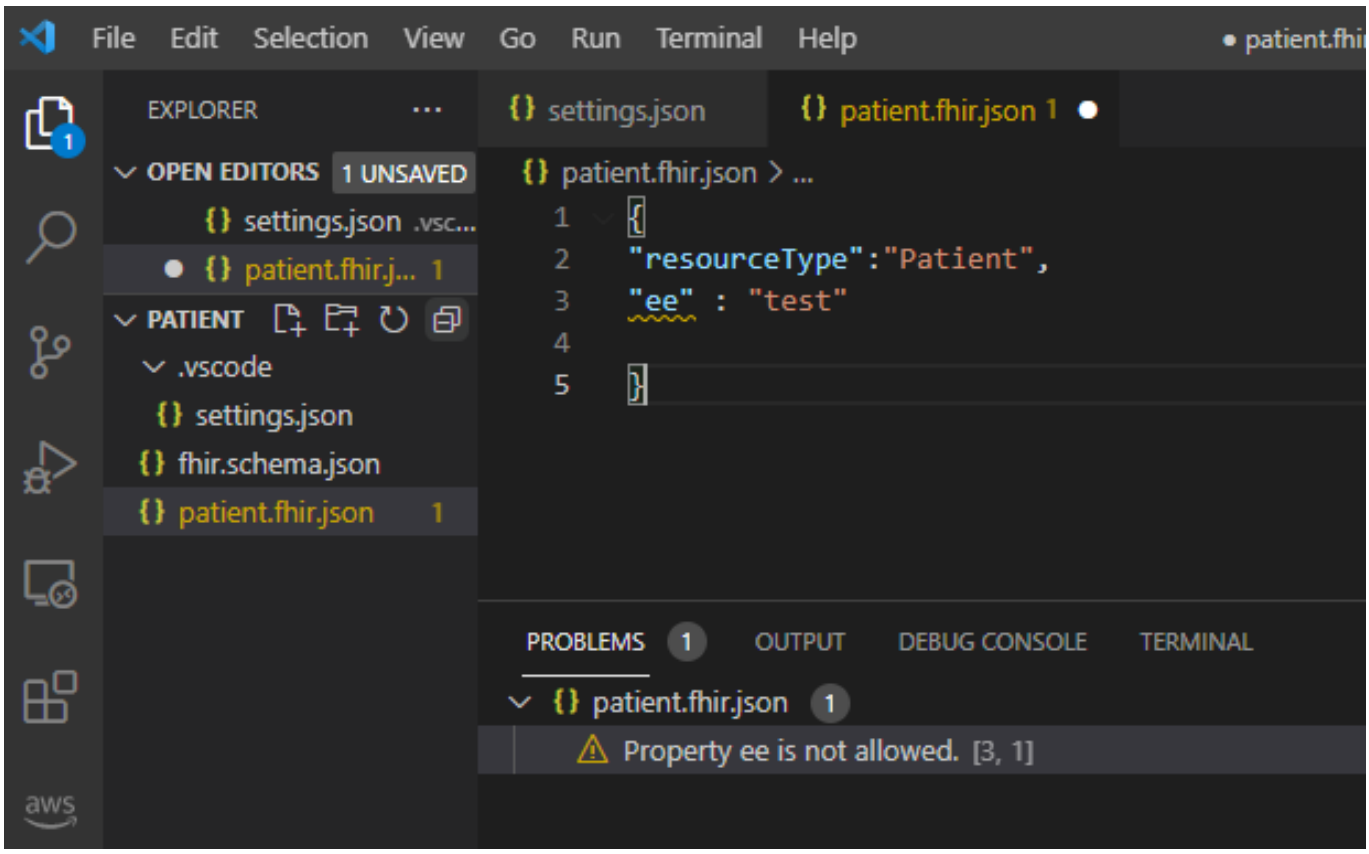
Paso 4: Crear un nuevo archivo patient.fhir.json, en la misma carpeta.
Pulsar Ctrl+ESPACIO y se obtendrán todos los atributos de los recursos de FHIR a través de IntelliSense



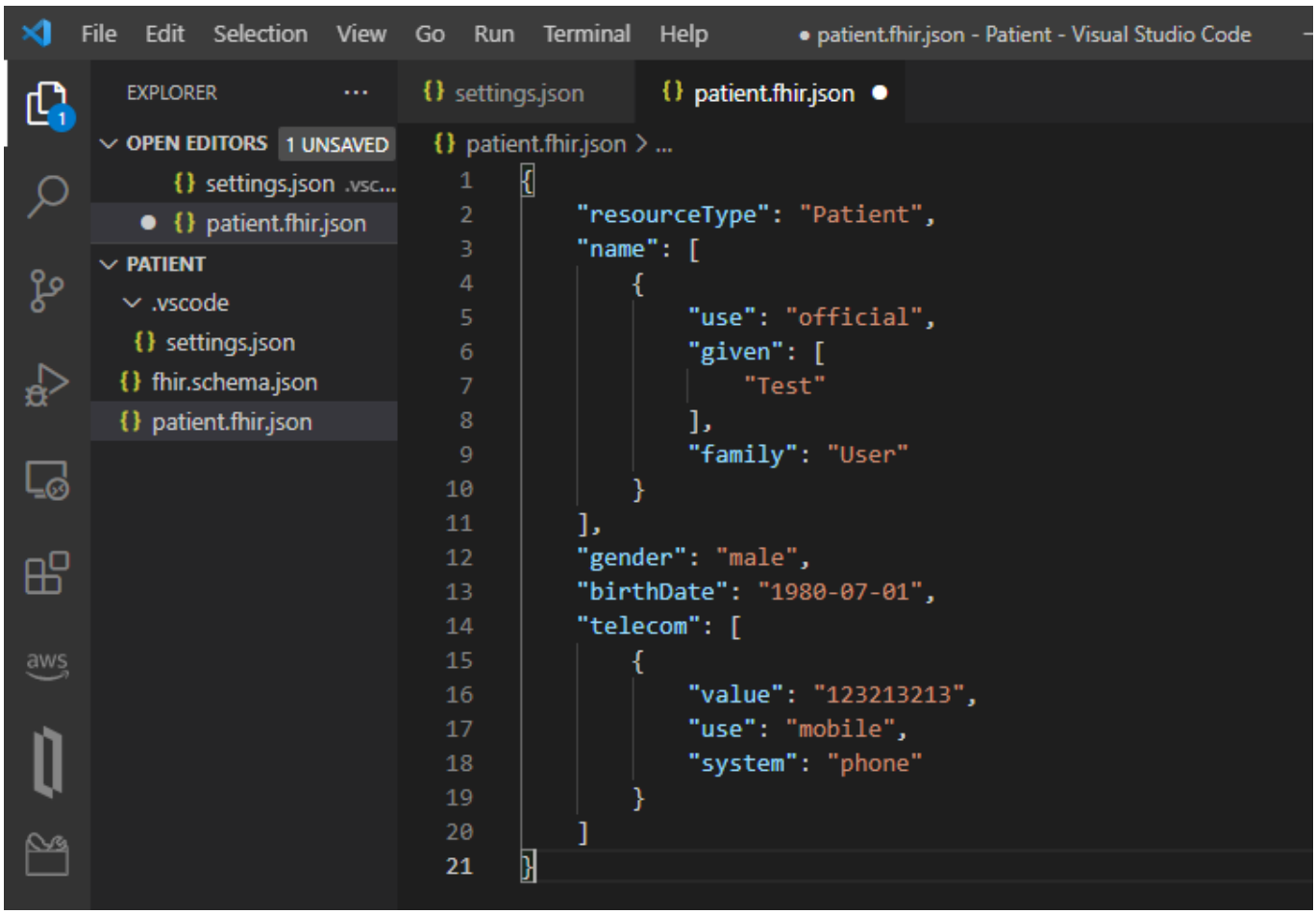
Escribir "resourceType": "Patient" y todos los atributos relacionados con el recurso Patient comenzarán a aparecer en el IntelliSense.



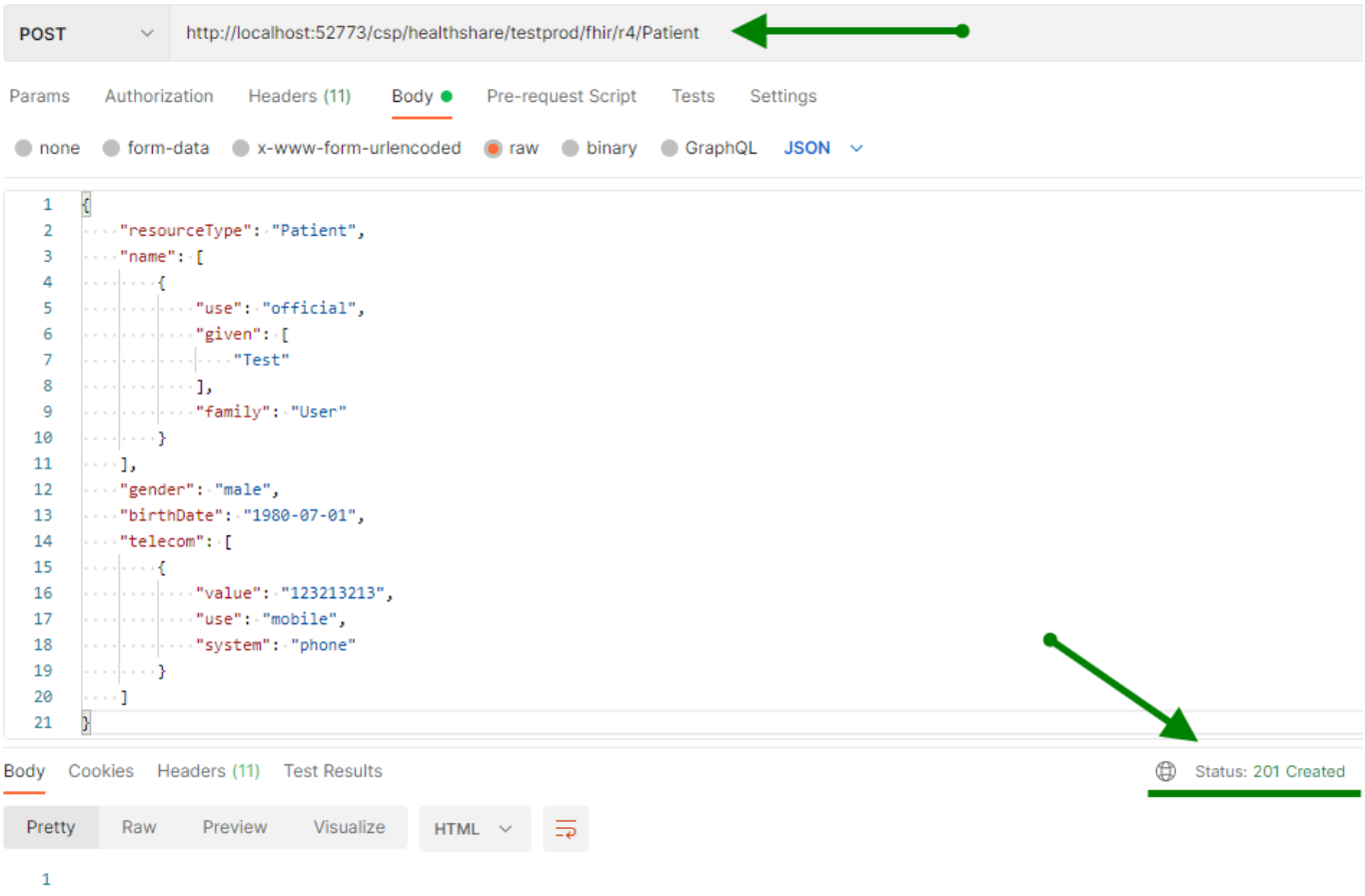
VS Code validará automáticamente la estructura y la sintaxis del recurso.



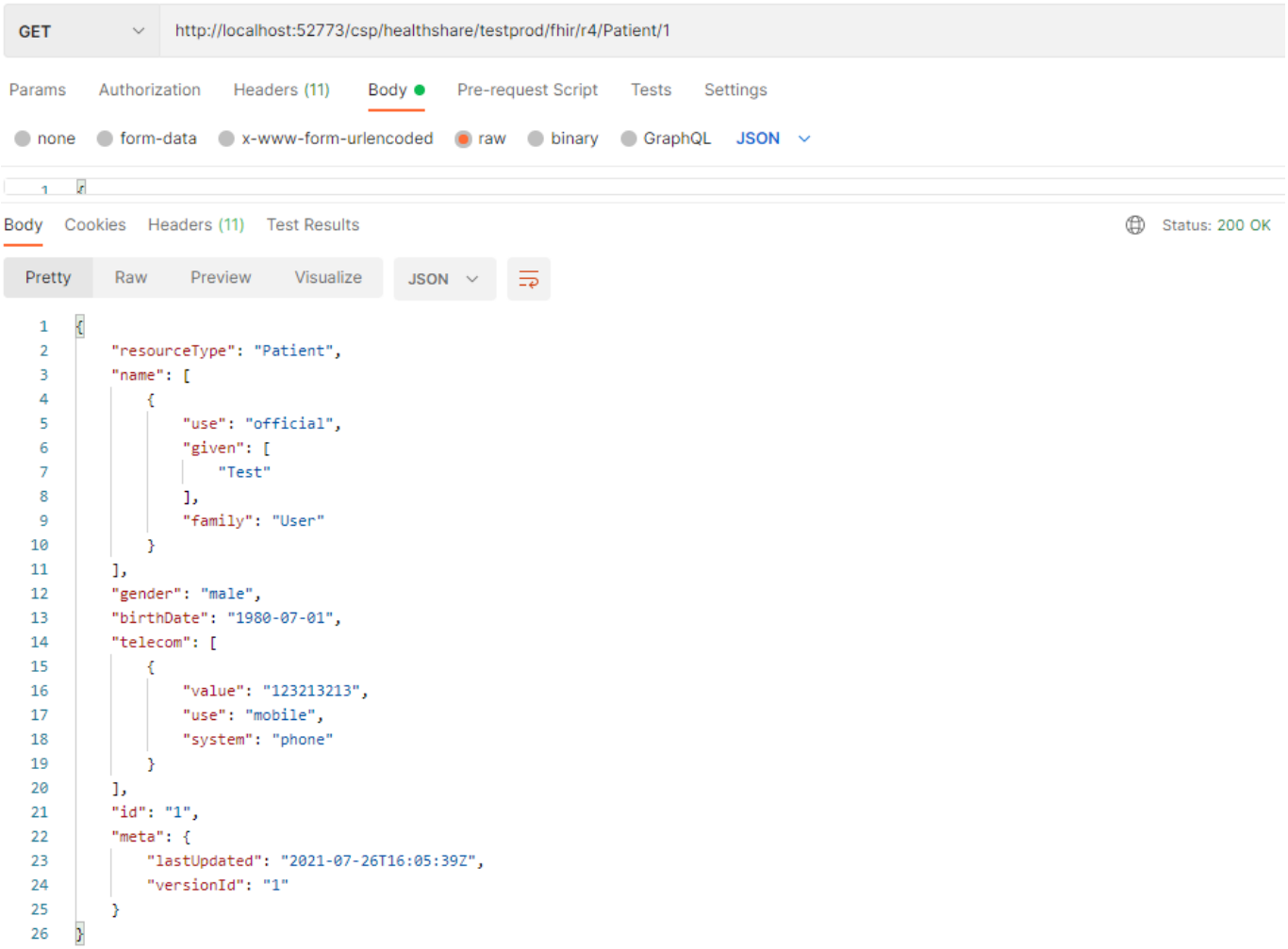
Con la ayuda de IntelliSense y la función de autocompletado hemos creado y validado nuestro recurso para los pacientes.



Paso 5: Publicar el recurso creado en el servidor FHIR de InterSystems usando la API Rest de Postman



Recuperar el recurso de Patient Creado utilizando el método GET



¡ Enhorabuena! Hemos creado y validado nuestro recurso Patient, y lo hemos publicado y recuperado correctamente en el Servidor FHIR de InterSystems usando Postman.

De esta manera podemos crear y validar fácilmente cualquier Recurso FHIR.

#API REST #FHIR #Caché #Ensemble #InterSystems IRIS for Health #VSCode

URL de fuente: https://es.community.intersystems.com/post/c%C3%B3mo-crear-y-validar-cualquier-recurso-fhir-usando-el-esquema-fhir-con-la-ayuda-de-intellisense