

**DRAFT: Web API Documentation for Spring Boot Final Project Weeks 3-6**<https://localhost:8080/>**(NOT YET IMPLEMENTED!!)**

Http Verb Method	Endpoint	Description
POST	/users	CREATE: Store a new user Needs firstName, lastName, userType, instruments, & address ( <b>creates a user &amp; address &amp; musician_instrument record</b> )
GET	/users	READ: Get all users — MUSICIANS & PLANNERS
PUT	/users/{id}	UPDATE: Update a user by {id} — Needs all fields
DELETE	/users/{id}	DELETE: Delete a customer by {id}
GET	/users/{id}/gigs	READ: Get all gigs that match user by {id}
PUT	/users/{id}/gigs/{id}/request	<b>UPDATE: Update a gig by id, adding user to instrument slot!</b>
POST	/instruments	CREATE: Stores an instrument — Needs a name. ( <b>creates an instrument record</b> )
GET	/instruments	READ: Get all instruments
PUT	/instruments/{id}	UPDATE: Update an instrument by {id} — Needs a name
DELETE	/instruments/{id}	DELETE: Delete an instrument by {id}
POST	/gigs	CREATE: Stores a gig — Needs date, phone, event, genre, plannerId description, salary ( <b>creates a gig record</b> )  <b>(DOESN'T ADD INSTRUMENTS YET — NEED TO DO THAT SEPARATELY!)</b>
GET	/gigs	READ: Gets all gigs

Http Verb Method	Endpoint	Description
GET	/gigs/{id}	READ: Get all instruments required for a gig by {id}. <i>(creates gig_status &amp; gig_status_instrument records)</i>
POST	/gigs/{id}	CREATE: Stores Instruments for a gig by {id} — <i>(creates a gig_status record)</i>
PUT	/gigs/{id}	UPDATE: Update a gig by {id}. <b>(FRAMEWORK IS THERE, NOT IMPLEMENTED YET!)</b>
DELETE	/gigs/{id}	DELETE: Delete a gig by {id}
GET	/gigs/{id}/users	READ: Get all users that match a particular gig by {id}
PUT	/gigs/{id}/users/{id}	<b>UPDATE: Update a gig {id} to “hire” a user by {id}</b>