

RabbitMQ producer/consumer keys:

routing_key in notifications amqp_setup.py : 'order.*'

Accounts

POST /accounts/login - Login	
Parameters	<pre>{ "user_email": "haniel@danley.com", "password": "TA" }</pre>
Responses	Content type – application/json
Code - 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "account_id": 1, "user_email": "haniel@danley.com", "user_phone": "99999999" }, "message": "Successfully logged in." }</pre>
Code - 404	<p>Description – no account with specified email</p> <p>Example Value</p> <pre>{ "message": "Unable to find account with specified email." }</pre>
Code - 403	<p>Description – wrong password</p> <p>Example Value</p> <pre>{ "message": "Wrong credentials. Please try again." }</pre>
GET /accounts – Get all accounts	
Parameters	No parameters
Responses	Content type – application/json

Hyperlinks: [Accounts](#) [Jobs](#) [Orders](#) [Place order](#) [Payment](#) [Stripe](#)

Code – 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "accounts": [{ "account_id": 1, "password": "sha256\$J7EfqFvkAAXbk15K\$e05cb9cb2cd5695316e65c6c4ac7455d381338e9f 2d1682f458cf333495e80fb", "user_email": "haniel@danley.com", "user_phone": "22222222" }, { "account_id": 2, "password": "sha256\$R6maOPGf4mokWN73\$833282503b64c197654e928908a7ca8cbb3f1 b77549301f249ed2992c10080e6", "user_email": "phris@coskitt.com", "user_phone": "99999999" }, { "account_id": 3, "password": "sha256\$n4bRJGZ7JI3pKGQm\$11592b4d112c92964f62b0d082c1c0d15672b69 48bb9ec04d22ff50a0f28776c", "user_email": "cposkitt@smu.edu.sg", "user_phone": "88888888" }] } }</pre>
------------	---

Code – 404	Description – no accounts Example Value <pre>{ "message": "There are no accounts." }</pre>
GET /accounts/<int:account_id> – Find account by account ID	
Parameters	
Name	Description
account_id int (path)	Unique ID of account to return Example Value 1
Responses	Content type – application/json
Code – 200	Description – successful operation Example Value <pre>{ "data": { "account_id": 1, "password": "sha256\$J7EfQFvkAAXbk15K\$e05cb9cb2cd5695316e65c6c4ac7455d381338e9f 2d1682f458cf333495e80fb", "user_email": "haniel@danley.com", "user_phone": "22222222" } }</pre>
Code – 404	Description – Account not found Example Value <pre>{ "message": "Account not found." }</pre>
POST /accounts – Create account	

Parameters	
Name	Description
body object (body)	<p>Account to create</p> <p>Example Value</p> <pre>{ "user_email": "sample@email.com", "password": "123456hola", "user_phone": "6778678" }</pre>
Responses	
Code – 201	<p>Content type – application/json</p> <p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "account_id": 4, "password": "sha256\$OTX7CEiu5Qh8JKSs\$ed8f984c5ab869a8c1608fa55c5b18a23f97b2a509fa4fb6e608a45d36cb2f32", "user_email": "sample@email.com", "user_phone": "6778678" } }</pre>
Code – 500	<p>Description – An error occurred creating the account</p> <p>Example Value</p> <pre>{ "message": "An error occurred creating the account.", "error": "Specific details of the error that occurred." }</pre>
PATCH /accounts/<int:account_id> – Update phone of account	
Parameters	
Name	Description

Hyperlinks: [Accounts](#) [Jobs](#) [Orders](#) [Place order](#) [Payment](#) [Stripe](#)

account_id int (path)	Unique ID of account to update Example Value 1
body object (body)	Account status to update Example Value <pre>{ "user_phone": "99999999" }</pre>
Responses	Content type – application/json
Code – 200	Description – successful operation Example Value <pre>{ "data": { "account_id": 1, "password": "sha256\$J7EfqFvkAAXbk15K\$e05cb9cb2cd5695316e65c6c4ac7455d381338e9f 2d1682f458cf333495e80fb", "user_email": "haniel@danley.com", "user_phone": "99999999" } }</pre>
Code – 404	Description – Account not found Example Value <pre>{ "data": { "account_id": 100 }, "message": "Account not found." }</pre>

Code – 500	<p>Description – An error occurred updating the account</p> <p>Example Value</p> <pre>{ "message": "An error occurred updating the account.", "error": "Specific details of the error that occurred." }</pre>
DELETE /accounts/<int:account_id> – Delete account	
Parameters	
Name	Description
account_id int (path)	<p>Unique ID of account to delete</p> <p>Example Value</p> <p>1</p>
Responses	Content type – application/json
Code – 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "account_id": 1 } }</pre>
Code – 404	<p>Description – Account not found</p> <p>Example Value</p> <pre>{ "data": { "account_id": 88888888 }, "message": "account not found." }</pre>
Code – 500	<p>Description – An error occurred deleting the account</p> <p>Example Value</p>

	<pre>{ "message": "An error occurred deleting the account.", "error": "Specific details of the error that occurred." }</pre>
--	--

Jobs

GET /jobs – Get all jobs	
Parameters	
Responses	Content type – application/json
Code – 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "jobs": [{ "description": "Design a professional website for your business.", "job_id": 1, "price": 40.0, "seller_id": 2, "title": "Design Portfolio Website" }, { "description": "Design a Logo for your business.", "job_id": 2, "price": 30.0, "seller_id": 2, "title": "Logo Design" }, { "description": "Hire my service to wreck any party you want to bail out from but forced to come.", "job_id": 3, "price": 60.5, "seller_id": 1, </pre>

	<pre> "title": "Party Wrecker" }, { "description": "Babysit your children at home.", "job_id": 4, "price": 70.0, "seller_id": 1, "title": "Babysit Children" }] } }</pre>
--	--

GET /jobs/<int:job_id> – Find job by job ID	
Parameters	
Name	Description
job_id int (path)	Unique ID of job to return Example Value 2
Responses	Content type – application/json
Code – 200	Description – successful operation Example Value <pre> { "data": { "description": "Design a Logo for your business.", "job_id": 2, "price": 30.0, "seller_id": 2, "title": "Logo Design" } }</pre>
Code – 404	Description – Job not found

	<p>Example Value</p> <pre>{ "message": "Job not found." }</pre>
GET /jobs/exclude/<int:seller_id> – Get all jobs which exclude those posted by the specific user	
Parameters	
Name	Description
seller_id int (path)	Unique ID of account not to return Example Value 1
Responses	Content type – application/json
Code – 200	Description – successful operation Example Value <pre>{ "data": { "jobs": [{ "description": "Design a professional website for your business.", "job_id": 1, "price": 40.0, "seller_id": 2, "title": "Design Portfolio Website" }, { "description": "Design a Logo for your business.", "job_id": 2, "price": 30.0, "seller_id": 2, "title": "Logo Design" }] } }</pre>

	<pre> } } </pre>
Code – 404	<p>Description – There is no jobs in the job database</p> <p>Example Value</p> <pre> { "message": "There are no jobs." } </pre>
POST /jobs – Create job	
Parameters	
Name	Description
body object (<i>body</i>)	<p>Job to create</p> <p>Example Value</p> <pre> { "price" : 100.0, "title" : "Walk your dog" , "description": "I will walk you dog for 100 bucks.", "seller_id": 3 } </pre>
Responses	Content type – application/json
Code – 201	<p>Description – successful operation</p> <p>Example Value</p> <pre> { "data": { "description": "I will walk you dog for 100 bucks.", "job_id": 29, "price": 100.0, "seller_id": 3, "title": "Walk your dog" } } </pre>
Code – 500	Description – An error occurred creating the job

	<p>Example Value</p> <pre>{ "message": "An error occurred creating the job.", "error": "Specific details of the error that occurred." }</pre>
PUT /jobs/<int:job_id> – Replace job	
Parameters	
Name	Description
job_id int <i>(path)</i>	<p>Unique ID of job to replace</p> <p>Example Value</p> <p>1</p>
body object <i>(body)</i>	<p>Entity to replace the existing one.</p> <p>Example Value</p> <pre>{ "price" : 2000, "title" : "Design Project Website", "description": "Will design your website.", "seller_id": 1 }</pre>
Responses	Content type – application/json
Code – 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "description": "Will design your website.", "job_id": 1, "price": 2000.0, "seller_id": 1, "title": "Design Project Website" } }</pre>

	}
Code – 404	Description – Job not found Example Value <pre>{ "data": { "job_id": 100 }, "message": "Job not found." }</pre>
Code – 500	Description – An error occurred replacing the job Example Value <pre>{ "message": "An error occurred replacing the job.", "error": "Specific details of the error that occurred." }</pre>
DELETE /jobs/<int:job_id> – Delete job	
Parameters	
Name	Description
job_id int (path)	Unique ID of job to delete Example Value 7
Responses	Content type – application/json
Code – 200	Description – successful operation Example Value <pre>{ "data": { "job_id": 1 } }</pre>

	}
Code – 404	<p>Description – Job not found</p> <p>Example Value</p> <pre>{ "data": { "job_id": 88888888 }, "message": "Job not found." }</pre>
Code – 500	<p>Description – An error occurred deleting the job</p> <p>Example Value</p> <pre>{ "message": "An error occurred deleting the job.", "error": "Specific details of the error that occurred." }</pre>
PATCH /jobs/<int:job_id> – Partially update job	
Parameters	
Name	Description
job_id int (path)	<p>Unique ID of job to update</p> <p>Example Value</p> <p>1</p>
body object (body)	<p>Values to update</p> <p>Example Value</p> <pre>{ "price" : 1999 }</pre>
Responses	Content type – application/json
Code – 200	<p>Description – successful operation</p> <p>Example Value</p>

	<pre>{ "data": { "description": "Design a professional website for your business.", "job_id": 1, "price": 1999.0, "seller_id": 1, "title": "Design Portfolio Website" } }</pre>
Code – 404	<p>Description – Job not found</p> <p>Example Value</p> <pre>{ "data": { "job_id": 88888888 }, "message": "Job not found." }</pre>
Code – 500	<p>Description – An error occurred updating the job</p> <p>Example Value</p> <pre>{ "message": "An error occurred updating the job.", "error": "Specific details of the error that occurred." }</pre>

Orders

GET /orders – Get all orders	
Parameters	No parameters
Responses	Content type – application/json
Code – 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": {</pre>

	<pre> "orders": [{ "created": "Tue, 10 Aug 2021 00:00:00 GMT", "customer_email": "buyer.cs302@gmail.com", "customer_phone": 99999999, "job_id": 1, "order_id": 5, "seller_id": 2, "status": "PENDING", "title": "Design Portfolio Website" }, { "created": "Tue, 10 Aug 2021 00:00:00 GMT", "customer_email": "seller.cs302@gmail.com", "customer_phone": 88888888, "job_id": 4, "order_id": 6, "seller_id": 1, "status": "PENDING", "title": "Babysit Children" }] } </pre>
Code – 404	<p>Description – no orders</p> <p>Example Value</p> <pre> { "message": "There are no orders." } </pre>
GET /orders/<int:order_id> – Find order by order ID	
Parameters	No parameters
Name	Description

order_id int (path)	Unique ID of order to return Example Value 6
Responses	Content type – application/json
Code – 200	Description – successful operation Example Value <pre>{ "data": { "created": "Tue, 10 Aug 2021 00:00:00 GMT", "customer_email": "seller.cs302@gmail.com", "customer_phone": 88888888, "job_id": 4, "order_id": 6, "seller_id": 1, "status": "PENDING", "title": "Babysit Children" } }</pre>
Code – 404	Description – Order not found Example Value <pre>{ "message": "Order not found." }</pre>
GET /orders/buyer/<str:email> – Find order by customer email	
Parameters	No parameters
Name	Description
email str (path)	Unique email of order to return Example Value "buyer.cs302@gmail.com"
Responses	Content type – application/json

Code – 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "orders": [{ "created": "Tue, 10 Aug 2021 00:00:00 GMT", "customer_email": "buyer.cs302@gmail.com", "customer_phone": 99999999, "job_id": 1, "order_id": 5, "seller_id": 2, "status": "PENDING", "title": "Design Portfolio Website" }] } }</pre>
Code – 404	<p>Description – Order not found</p> <p>Example Value</p> <pre>{ "message": "Order not found." }</pre>
GET /orders/seller/<int:account_id> – Find order by seller account ID	
Parameters	No parameters
Name	Description
account_id int (path)	<p>Unique ID of account to return</p> <p>Example Value</p> <p>1</p>
Responses	Content type – application/json

Code – 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "orders": [{ "created": "Tue, 10 Aug 2021 00:00:00 GMT", "customer_email": "seller.cs302@gmail.com", "customer_phone": 888888888, "job_id": 4, "order_id": 6, "seller_id": 1, "status": "PENDING", "title": "Babysit Children" }] } }</pre>
Code – 404	<p>Description – Order not found</p> <p>Example Value</p> <pre>{ "message": "Order not found." }</pre>
POST /orders – Create order	
Parameters	
Name	Description
body object (<i>body</i>)	<p>Order to create</p> <p>Example Value</p> <pre>{ "customer_email": "haniel@danley.com",</pre>

	<pre> "customer_phone": 22222222, "seller_id": 1, "job_id": 9, "title": "Free Therapy" } </pre>
Responses	Content type – application/json
Code – 201	<p>Description – successful operation</p> <p>Example Value</p> <pre> { "data": { "created": "Thu, 11 Nov 2021 04:07:21 GMT", "customer_email": "haniel@danley.com", "customer_phone": 22222222, "job_id": 9, "order_id": 7, "seller_id": 1, "status": "PENDING", "title": "Free Therapy" } } </pre>
Code – 500	<p>Description – An error occurred creating the order</p> <p>Example Value</p> <pre> { "message": "An error occurred creating the order.", "error": "Specific details of the error that occurred." } </pre>
PATCH /orders/<int:order_id> – Update status of order	
Parameters	
Name	Description
order_id	Unique ID of order to update
int	
(path)	Example Value

	6
body object (body)	Order status to update Example Value <pre>{ "status": "COMPLETED" }</pre>
Responses	Content type – application/json
Code – 200	Description – successful operation Example Value <pre>{ "data": { "created": "Tue, 10 Aug 2021 00:00:00 GMT", "customer_email": "seller.cs302@gmail.com", "customer_phone": 88888888, "job_id": 4, "order_id": 6, "seller_id": 1, "status": "COMPLETED", "title": "Babysit Children" } }</pre>
Code – 404	Description – Order not found Example Value <pre>{ "data": { "order_id": 88888888 }, "message": "Order not found." }</pre>
Code – 500	Description – An error occurred updating the order

	<p>Example Value</p> <pre>{ "message": "An error occurred updating the order.", "error": "Specific details of the error that occurred." }</pre>
--	---

Place order

POST /place-order – Place an order based on customer’s job selection	
Parameters	
Name	Description
body object (<i>body</i>)	<p>Order to create. Notification will be sent to the AMQP broker only if order can be created.</p> <p>Example Value</p> <pre>{ "customer_email": "buyer.cs302@gmail.com", "customer_phone": "99999999", "job_id": 6, "title": "Logo Design", "seller_id": 2, "notifyemail": "seller.cs302@gmail.com", "notiinfo": "Customer buyer.cs302@gmail.com placed an order on Your job Logo Design" }</pre>
Responses	Content type – application/json

Code – 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "created": "Fri, 12 Nov 2021 01:50:31 GMT", "customer_email": "buyer.cs302@gmail.com", "customer_phone": 99999999, "job_id": 6, "order_id": 9, "seller_id": 2, "status": "PENDING", "title": "Logo Design" }, "message": "Order placed." }</pre>
Code – 500	<p>Description – Unable to create order record</p> <p>Example Value</p> <pre>{ "message": "Unable to place order.", "error": "Unable to create order record." }</pre>

PATCH /place-order/update-status

– Update the status of the order with order id and new status specified in the body. A notification will be sent to the email address specified in the body.

Parameters	
Name	Description
body object (body)	order id of the order to be updated and the new status Example Value { "status": "COMPLETED", "order_id": 5, "notifyemail": "buyer.cs302@gmail.com", "notiinfo": "Seller seller.cs302@gmail.com completed your order Logo Design" }
Responses	Content type – application/json

Code – 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "created": "Tue, 10 Aug 2021 00:00:00 GMT", "customer_email": "buyer.cs302@gmail.com", "customer_phone": 99999999, "job_id": 1, "order_id": 5, "seller_id": 2, "status": "COMPLETED", "title": "Design Portfolio Website" }, "message": "Order status updated." }</pre>
------------	---

Payment

POST / – Receive payment details	
Parameters	
Name	Description
body object (<i>body</i>)	<p>Payment details</p> <p>Example Value</p> <pre>{ "username": "test", "email": "test@gmail.com", "amount": 2000, "order_id": "5" }</pre>
Responses	Content type – application/json
Code – 200	Description – successful operation

	<p>Example Value</p> <pre>{ "username": "test", "email": "test@gmail.com", "amount": 2000, "order_id": "5" }</pre>
Code – 500	<p>Description – An error occurred receiving payment details</p> <p>Example Value</p> <pre>{ "message": "An error occurred receiving payment details.", "error": "Error message from caught exception" }</pre>

POST /webhook – Called after Stripe PaymentIntent succeeds	
Parameters	
Name	Description
body object (body)	<p>Payment details</p> <p>Example Value</p> <p>Event response from Stripe. Omitted for brevity.</p>
Responses	Content type – application/json
Code – 200	<p>Description – successful operation</p> <p>Example Value</p> <pre>{ "data": { "created": "Tue, 10 Aug 2021 00:00:00 GMT", "customer_email": "cposkitt@smu.edu.sg", "customer_phone": 88888888, "job_id": 2, "order_id": 5, "seller_id": 1, "status": "PAID", "title": "Party Wrecker" } }</pre>

	200
Code – 500	Description – An error occurred when calling the webhook 500

GET /check-stripe – Check connection to Stripe service	
Parameters	None
Responses	Content type – string
Code – 200	Description – successful operation Example Value <pre>{ 'message': 'Service is healthy' }</pre>
Code – 200	Description – There is no connection to Stripe Example Value "" (empty String)

GET /health – Check health of payment service	
Parameters	None
Responses	Content type – string
Code – 200	Description – successful operation Example Value <pre>{ 'message': 'Service is healthy' }</pre>
Code – None (ECONNREFUSED)	Description – The payment service is not running / cannot be contacted Example Value

GET /get-active-profile – Check current active profile

Parameters	None
Responses	Content type – string
Code – 200	Description – successful operation Example Value dev
Code – 200	Description – There is no active profile set Example Value Unknown

Stripe

POST /create-payment-intent – Creates Stripe PaymentIntent	
Parameters	
Name	Description
body object (<i>body</i>)	Payment details Example Value <pre>{ "username": "test", "email": "test@gmail.com", "amount": 2000, "order_id": "5" }</pre>
Responses	Content type – String
Code – 200	Description – successful operation Example Value <pre>{ "clientSecret": <Stripe client secret> }</pre>
Code – 400	Description – A StripeException was caught Example Value <pre>{ "error": <StripeException error data>, }</pre>

	<pre>"status": 400 }</pre>
Code – 500	<p>Description – A generic exception was caught</p> <p>Example Value</p> <pre>{ "error": <Exception error data>, "status": 500 }</pre>

POST / – Receives payment details for UI	
Parameters	
Name	Description
body object (body)	<p>Payment details</p> <p>Example Value</p> <pre>{ "username": "test", "email": "test@gmail.com", "amount": 2000, "order_id": "5" }</pre>
Responses	Content type – void
Code – 200	Description – successful operation
Code – 500	Description – There is no connection to the Stripe service

GET /check-payment – Check connection to Payment service	
Parameters	None
Responses	Content type – string
Code – 200	<p>Description – successful operation</p> <p>Example Value</p>

Hyperlinks: [Accounts](#) [Jobs](#) [Orders](#) [Place order](#) [Payment](#) [Stripe](#)

	<pre>{ 'message': 'Service is healthy' }</pre>
Code – 200	Description – There is no connection to Payment Example Value null

GET /config – Obtain Stripe publishable key	
Parameters	None
Responses	Content type – string
Code – 200	Description – successful operation Example Value <pre>{ 'publishableKey': <Stripe publishable key> }</pre>
Code – 200	Description – There is no publishable key value Example Value <pre>{ 'publishableKey': null }</pre>

GET /health – Check health of Stripe service	
Parameters	None
Responses	Content type – string
Code – 200	Description – successful operation Example Value <pre>{ 'message': 'Stripe service is healthy' }</pre>

Code – None (ECONNREFUSED)	Description – The service is not running / cannot be contacted Example Value
-------------------------------	---

GET /get-active-profile – Check current active profile	
Parameters	None
Responses	Content type – string
Code – 200	Description – successful operation Example Value dev
Code – 200	Description – There is no active profile set Example Value Unknown