



```
1 {
2   "openapi": "3.0.3",
3   "info": {
4     "title": "Api Documentation",
5     "description": "Api Documentation",
6     "termsOfService": "urn:tos",
7     "contact": {},
8     "license": {
9       "name": "Apache 2.0",
10      "url": "http://www.apache.org/licenses/LICENSE-2.0"
11    },
12    "version": "1.0"
13  },
14  "servers": [
15    {
16      "url": "http://localhost:8080",
17      "description": "Inferred Url"
18    }
19  ],
20  "tags": [
21    {
22      "name": "basic-error-controller",
23      "description": "Basic Error Controller"
24    },
25    {
26      "name": "restaurant-controller",
27      "description": "REST API's related to restaurants and receiving information from
28 Yelp's API about them"
29    },
30    {
31      "name": "user-controller",
32      "description": "User Controller"
33    }
34  ],
35  "paths": {
36    "/error": {
37      "get": {
38        "tags": [
39          "basic-error-controller"
40        ],
41        "summary": "errorHtml",
42        "operationId": "errorHtmlUsingGET",
43        "responses": {
44          "200": {
45            "description": "OK",
46            "content": {
47              "text/html": {
48                "schema": {
49                  "$ref": "#/components/schemas/ModelAndView"
50                }
51              }
52            }
53          },
54          "401": {
55            "description": "Unauthorized"
56          },
57          "403": {
58            "description": "Forbidden"
59          },
60          "404": {
61            "description": "Not Found"
62          }
63        }
64      },
65      "put": {
66        "tags": [
67          "basic-error-controller"
68        ],
69        "summary": "errorHtml",
70        "operationId": "errorHtmlUsingPUT",
71        "responses": {
72          "200": {
73            "description": "OK",
74            "content": {
```

```

75         "text/html": {
76             "schema": {
77                 "$ref": "#/components/schemas/ModelAndView"
78             }
79         }
80     },
81     "201": {
82         "description": "Created"
83     },
84     "401": {
85         "description": "Unauthorized"
86     },
87     "403": {
88         "description": "Forbidden"
89     },
90     "404": {
91         "description": "Not Found"
92     }
93 },
94 },
95 },
96 "post": {
97     "tags": [
98         "basic-error-controller"
99     ],
100     "summary": "errorHtml",
101     "operationId": "errorHtmlUsingPOST",
102     "responses": {
103         "200": {
104             "description": "OK",
105             "content": {
106                 "text/html": {
107                     "schema": {
108                         "$ref": "#/components/schemas/ModelAndView"
109                     }
110                 }
111             }
112         },
113         "201": {
114             "description": "Created"
115         },
116         "401": {
117             "description": "Unauthorized"
118         },
119         "403": {
120             "description": "Forbidden"
121         },
122         "404": {
123             "description": "Not Found"
124         }
125     }
126 },
127 "delete": {
128     "tags": [
129         "basic-error-controller"
130     ],
131     "summary": "errorHtml",
132     "operationId": "errorHtmlUsingDELETE",
133     "responses": {
134         "200": {
135             "description": "OK",
136             "content": {
137                 "text/html": {
138                     "schema": {
139                         "$ref": "#/components/schemas/ModelAndView"
140                     }
141                 }
142             }
143         },
144         "204": {
145             "description": "No Content"
146         },
147         "401": {
148             "description": "Unauthorized"

```

```

149         },
150         "403": {
151             "description": "Forbidden"
152         }
153     },
154     },
155     "options": {
156         "tags": [
157             "basic-error-controller"
158         ],
159         "summary": "errorHtml",
160         "operationId": "errorHtmlUsingOPTIONS",
161         "responses": {
162             "200": {
163                 "description": "OK",
164                 "content": {
165                     "text/html": {
166                         "schema": {
167                             "$ref": "#/components/schemas/ModelAndView"
168                         }
169                     }
170                 }
171             },
172             "204": {
173                 "description": "No Content"
174             },
175             "401": {
176                 "description": "Unauthorized"
177             },
178             "403": {
179                 "description": "Forbidden"
180             }
181         }
182     },
183     "head": {
184         "tags": [
185             "basic-error-controller"
186         ],
187         "summary": "errorHtml",
188         "operationId": "errorHtmlUsingHEAD",
189         "responses": {
190             "200": {
191                 "description": "OK",
192                 "content": {
193                     "text/html": {
194                         "schema": {
195                             "$ref": "#/components/schemas/ModelAndView"
196                         }
197                     }
198                 }
199             },
200             "204": {
201                 "description": "No Content"
202             },
203             "401": {
204                 "description": "Unauthorized"
205             },
206             "403": {
207                 "description": "Forbidden"
208             }
209         }
210     },
211     "patch": {
212         "tags": [
213             "basic-error-controller"
214         ],
215         "summary": "errorHtml",
216         "operationId": "errorHtmlUsingPATCH",
217         "responses": {
218             "200": {
219                 "description": "OK",
220                 "content": {
221                     "text/html": {
222                         "schema": {

```

```

223         "$ref": "#/components/schemas/ModelAndView"
224     }
225 }
226 }
227 },
228 "204": {
229     "description": "No Content"
230 },
231 "401": {
232     "description": "Unauthorized"
233 },
234 "403": {
235     "description": "Forbidden"
236 }
237 }
238 },
239 "trace": {
240     "tags": [
241         "basic-error-controller"
242     ],
243     "summary": "errorHtml",
244     "operationId": "errorHtmlUsingTRACE",
245     "responses": {
246         "200": {
247             "description": "OK",
248             "content": {
249                 "text/html": {
250                     "schema": {
251                         "$ref": "#/components/schemas/ModelAndView"
252                     }
253                 }
254             }
255         },
256         "204": {
257             "description": "No Content"
258         },
259         "401": {
260             "description": "Unauthorized"
261         },
262         "403": {
263             "description": "Forbidden"
264         }
265     }
266 }
267 },
268 "/home/{city}/{price}/{categories}/{attributes}": {
269     "get": {
270         "tags": [
271             "CustomURLErrorController"
272         ],
273         "summary": "Allows the user to completely customize the restaurants they see.
274 They can update their location, radius, update their accommodations and preferences, will auto
275 update the url based on the user's wants",
276         "operationId": "getSwipingUsingGET",
277         "parameters": [
278             {
279                 "name": "city",
280                 "in": "path",
281                 "description": "city",
282                 "required": true,
283                 "style": "simple",
284                 "schema": {
285                     "type": "string"
286                 }
287             },
288             {
289                 "name": "price",
290                 "in": "path",
291                 "description": "price",
292                 "required": true,
293                 "style": "simple",
294                 "schema": {
295                     "type": "integer",
296                     "format": "int32"

```

```

296     }
297   },
298   {
299     "name": "categories",
300     "in": "path",
301     "description": "categories",
302     "required": true,
303     "style": "simple",
304     "schema": {
305       "type": "string"
306     }
307   },
308   {
309     "name": "attributes",
310     "in": "path",
311     "description": "attributes",
312     "required": true,
313     "style": "simple",
314     "schema": {
315       "type": "string"
316     }
317   }
318 ],
319 "responses": {
320   "200": {
321     "description": "Success!",
322     "content": {
323       "*/*": {
324         "schema": {
325           "type": "array",
326           "items": {
327             "$ref": "#/components/schemas/JsonNode"
328           }
329         }
330       }
331     }
332   },
333   "401": {
334     "description": "Not authorized!"
335   },
336   "403": {
337     "description": "Forbidden!"
338   },
339   "404": {
340     "description": "Not found"
341   },
342   "510": {
343     "description": "Internal server error"
344   }
345 }
346 },
347 "/restaurant/find/{code}": {
348   "get": {
349     "tags": [
350       "FindRestaurantByCode"
351     ],
352     "summary": "Find a specific restaurant by their unique code",
353     "operationId": "findRestaurantByCodeUsingGET",
354     "parameters": [
355       {
356         "name": "code",
357         "in": "path",
358         "description": "code",
359         "required": true,
360         "style": "simple",
361         "schema": {
362           "type": "string"
363         }
364       }
365     ],
366     "responses": {
367       "200": {
368         "description": "Success!",

```

```

370         "content": {
371             "*/*": {
372                 "schema": {
373                     "$ref": "#/components/schemas/JsonNode"
374                 }
375             }
376         },
377     },
378     "401": {
379         "description": "Not authorized!"
380     },
381     "403": {
382         "description": "Forbidden!"
383     },
384     "404": {
385         "description": "Not found"
386     },
387     "510": {
388         "description": "Internal server error"
389     }
390 },
391 },
392 },
393 "/restaurant/new-restaurant": {
394     "put": {
395         "tags": [
396             "CreateRestaurant"
397         ],
398         "summary": "Create a new restaurant object",
399         "operationId": "createRestaurantUsingPUT",
400         "requestBody": {
401             "content": {
402                 "application/json": {
403                     "schema": {
404                         "$ref": "#/components/schemas/RestaurantReq"
405                     }
406                 }
407             }
408         },
409         "responses": {
410             "200": {
411                 "description": "Success!",
412                 "content": {
413                     "*/*": {
414                         "schema": {
415                             "type": "string"
416                         }
417                     }
418                 }
419             },
420             "201": {
421                 "description": "Created"
422             },
423             "401": {
424                 "description": "Not authorized!"
425             },
426             "403": {
427                 "description": "Forbidden!"
428             },
429             "404": {
430                 "description": "Not found"
431             },
432             "510": {
433                 "description": "Internal server error"
434             }
435         }
436     }
437 },
438 "/restaurant/reviews/{code}": {
439     "get": {
440         "tags": [
441             "GetRestaurantReviews"
442         ],
443         "summary": "Find a specific restaurant's reviews, limit is 3",

```

```

444     "operationId": "getRestaurantByReviewsUsingGET",
445     "parameters": [
446         {
447             "name": "code",
448             "in": "path",
449             "description": "code",
450             "required": true,
451             "style": "simple",
452             "schema": {
453                 "type": "string"
454             }
455         }
456     ],
457     "responses": {
458         "200": {
459             "description": "Success!",
460             "content": {
461                 "*/*": {
462                     "schema": {
463                         "$ref": "#/components/schemas/JsonNode"
464                     }
465                 }
466             }
467         },
468         "401": {
469             "description": "Not authorized!"
470         },
471         "403": {
472             "description": "Forbidden!"
473         },
474         "404": {
475             "description": "Not found"
476         },
477         "510": {
478             "description": "Internal server error"
479         }
480     }
481 },
482 "/restaurant/{city}/all": {
483     "get": {
484         "tags": [
485             "getRestaurantsInCity"
486         ],
487         "summary": "Get all restaurants in the user-specified city, limit is 50",
488         "operationId": "getAllRestaurantsUsingGET",
489         "parameters": [
490             {
491                 "name": "city",
492                 "in": "path",
493                 "description": "city",
494                 "required": true,
495                 "style": "simple",
496                 "schema": {
497                     "type": "string"
498                 }
499             }
500         ],
501         "responses": {
502             "200": {
503                 "description": "Success!",
504                 "content": {
505                     "*/*": {
506                         "schema": {
507                             "type": "array",
508                             "items": {
509                                 "$ref": "#/components/schemas/JsonNode"
510                             }
511                         }
512                     }
513                 }
514             },
515             "401": {
516                 "description": "Not authorized!"
517             }
518         }
519     }
520 }

```

```

518         },
519         "403": {
520             "description": "Forbidden!"
521         },
522         "404": {
523             "description": "Not found"
524         },
525         "510": {
526             "description": "Internal server error"
527         }
528     }
529 },
530 "/restaurant/{code}": {
531     "put": {
532         "tags": [
533             "UpdateRestaurant"
534         ],
535         "summary": "Update a restaurant object",
536         "operationId": "updateRestaurantUsingPUT",
537         "parameters": [
538             {
539                 "name": "code",
540                 "in": "path",
541                 "description": "code",
542                 "required": true,
543                 "style": "simple",
544                 "schema": {
545                     "type": "string"
546                 }
547             }
548         ],
549         "requestBody": {
550             "content": {
551                 "application/json": {
552                     "schema": {
553                         "$ref": "#/components/schemas/RestaurantReq"
554                     }
555                 }
556             }
557         },
558         "responses": {
559             "200": {
560                 "description": "Success!",
561                 "content": {
562                     "*/*": {
563                         "schema": {
564                             "$ref": "#/components/schemas/RestaurantRes"
565                         }
566                     }
567                 }
568             },
569             "201": {
570                 "description": "Created"
571             },
572             "401": {
573                 "description": "Not authorized!"
574             },
575             "403": {
576                 "description": "Forbidden!"
577             },
578             "404": {
579                 "description": "Not found"
580             },
581             "510": {
582                 "description": "Internal server error"
583             }
584         }
585     },
586     "delete": {
587         "tags": [
588             "DeleteRestaurant"
589         ],
590         "summary": "Delete a restaurant object from the database",
591

```



```

592     "operationId": "deleteRestaurantUsingDELETE",
593     "parameters": [
594         {
595             "name": "code",
596             "in": "path",
597             "description": "code",
598             "required": true,
599             "style": "simple",
600             "schema": {
601                 "type": "string"
602             }
603         }
604     ],
605     "responses": {
606         "200": {
607             "description": "Success!",
608             "content": {
609                 "*/*": {
610                     "schema": {
611                         "type": "string"
612                     }
613                 }
614             }
615         },
616         "204": {
617             "description": "No Content"
618         },
619         "401": {
620             "description": "Not authorized!"
621         },
622         "403": {
623             "description": "Forbidden!"
624         },
625         "404": {
626             "description": "Not found"
627         },
628         "510": {
629             "description": "Internal server error"
630         }
631     }
632 },
633 {
634     "/friend/addFriend/{sent}": {
635         "post": {
636             "tags": [
637                 "AddNewFriend"
638             ],
639             "summary": "Adds a new friend to the given user",
640             "operationId": "addNewFriendUsingPOST",
641             "parameters": [
642                 {
643                     "name": "sent",
644                     "in": "path",
645                     "description": "sent",
646                     "required": true,
647                     "style": "simple",
648                     "schema": {
649                         "type": "string"
650                     }
651                 }
652             ],
653             "responses": {
654                 "200": {
655                     "description": "Success!",
656                     "content": {
657                         "*/*": {
658                             "schema": {
659                                 "type": "string"
660                             }
661                         }
662                     }
663                 },
664                 "201": {
665                     "description": "Created"

```

```

666         },
667         "401": {
668             "description": "Not authorized!"
669         },
670         "403": {
671             "description": "Forbidden!"
672         },
673         "404": {
674             "description": "Not found"
675         },
676         "510": {
677             "description": "Internal server error"
678         }
679     }
680 },
681 },
682 "/friend/{username}/find/{friend}": {
683     "get": {
684         "tags": [
685             "FindFriendByUsername"
686         ],
687         "summary": "Find a specific friend by their username",
688         "operationId": "findFriendByUsernameUsingGET",
689         "parameters": [
690             {
691                 "name": "username",
692                 "in": "path",
693                 "description": "username",
694                 "required": true,
695                 "style": "simple",
696                 "schema": {
697                     "type": "string"
698                 }
699             },
700             {
701                 "name": "friend",
702                 "in": "path",
703                 "description": "friend",
704                 "required": true,
705                 "style": "simple",
706                 "schema": {
707                     "type": "string"
708                 }
709             }
710         ],
711         "responses": {
712             "200": {
713                 "description": "Success!",
714                 "content": {
715                     "*/*": {
716                         "schema": {
717                             "$ref": "#/components/schemas/UserRes"
718                         }
719                     }
720                 }
721             },
722             "401": {
723                 "description": "Not authorized!"
724             },
725             "403": {
726                 "description": "Forbidden!"
727             },
728             "404": {
729                 "description": "Not found"
730             },
731             "510": {
732                 "description": "Internal server error"
733             }
734         }
735     }
736 },
737 "/friend/{username}/getAll": {
738     "get": {
739         "tags": [

```

```

740         "GetAllFriends"
741     ],
742     "summary": "Returns all friends the user has",
743     "operationId": "getAllFriendsUsingGET",
744     "parameters": [
745         {
746             "name": "username",
747             "in": "path",
748             "description": "username",
749             "required": true,
750             "style": "simple",
751             "schema": {
752                 "type": "string"
753             }
754         }
755     ],
756     "responses": {
757         "200": {
758             "description": "Success!",
759             "content": {
760                 "*/*": {
761                     "schema": {
762                         "uniqueItems": true,
763                         "type": "array",
764                         "items": {
765                             "type": "string"
766                         }
767                     }
768                 }
769             }
770         },
771         "401": {
772             "description": "Not authorized!"
773         },
774         "403": {
775             "description": "Forbidden!"
776         },
777         "404": {
778             "description": "Not found"
779         },
780         "510": {
781             "description": "Internal server error"
782         }
783     }
784 },
785 {
786     "/friend/{username}/remove/{friendUsername}": {
787         "put": {
788             "tags": [
789                 "RemoveFriend"
790             ],
791             "summary": "Remove a friend by their username",
792             "operationId": "removeFriendUsingPUT",
793             "parameters": [
794                 {
795                     "name": "username",
796                     "in": "path",
797                     "description": "username",
798                     "required": true,
799                     "style": "simple",
800                     "schema": {
801                         "type": "string"
802                     }
803                 },
804                 {
805                     "name": "friendUsername",
806                     "in": "path",
807                     "description": "friendUsername",
808                     "required": true,
809                     "style": "simple",
810                     "schema": {
811                         "type": "string"
812                     }
813                 }

```

```

814     ],
815     "responses": {
816         "200": {
817             "description": "Success!",
818             "content": {
819                 "*/*": {
820                     "schema": {
821                         "type": "string"
822                     }
823                 }
824             }
825         },
826         "201": {
827             "description": "Created"
828         },
829         "401": {
830             "description": "Not authorized!"
831         },
832         "403": {
833             "description": "Forbidden!"
834         },
835         "404": {
836             "description": "Not found"
837         },
838         "510": {
839             "description": "Internal server error"
840         }
841     }
842 },
843 {
844     "/users": {
845         "get": {
846             "tags": [
847                 "GetAllUsers"
848             ],
849             "summary": "Get all users from the database",
850             "operationId": "getAllUsersUsingGET",
851             "responses": {
852                 "200": {
853                     "description": "Success!",
854                     "content": {
855                         "*/*": {
856                             "schema": {
857                                 "type": "array",
858                                 "items": {
859                                     "$ref": "#/components/schemas/UserRes"
860                                 }
861                             }
862                         }
863                     }
864                 },
865                 "401": {
866                     "description": "Not authorized!"
867                 },
868                 "403": {
869                     "description": "Forbidden!"
870                 },
871                 "404": {
872                     "description": "Not found"
873                 },
874                 "510": {
875                     "description": "Internal server error"
876                 }
877             }
878         },
879         "post": {
880             "tags": [
881                 "CreateUsers"
882             ],
883             "summary": "Create new user",
884             "operationId": "createUserUsingPOST",
885             "requestBody": {
886                 "content": {
887                     "application/json": {

```

```

888         "schema": {
889             "$ref": "#/components/schemas/UserReq"
890         }
891     }
892 },
893 "responses": {
894     "200": {
895         "description": "Success!",
896         "content": {
897             "*/*": {
898                 "schema": {
899                     "type": "string"
900                 }
901             }
902         }
903     },
904     "201": {
905         "description": "Created"
906     },
907     "401": {
908         "description": "Not authorized!"
909     },
910     "403": {
911         "description": "Forbidden!"
912     },
913     "404": {
914         "description": "Not found"
915     },
916     "510": {
917         "description": "Internal server error"
918     }
919 },
920 },
921 },
922 },
923 "/users/login/{sent}": {
924     "get": {
925         "tags": [
926             "GetUserByUsername"
927         ],
928         "summary": "Get a specific user by username",
929         "operationId": "getUserByUsernameUsingGET",
930         "parameters": [
931             {
932                 "name": "sent",
933                 "in": "path",
934                 "description": "sent",
935                 "required": true,
936                 "style": "simple",
937                 "schema": {
938                     "type": "string"
939                 }
940             }
941         ],
942         "responses": {
943             "200": {
944                 "description": "Success!",
945                 "content": {
946                     "*/*": {
947                         "schema": {
948                             "type": "object"
949                         }
950                     }
951                 }
952             },
953             "401": {
954                 "description": "Not authorized!"
955             },
956             "403": {
957                 "description": "Forbidden!"
958             },
959             "404": {
960                 "description": "Not found"
961             }
962         }
963     }
964 }

```

```

962         "510": {
963             "description": "Internal server error"
964         }
965     }
966 }
967 },
968 "/users/{id}": {
969     "get": {
970         "tags": [
971             "GetUserById"
972         ],
973         "summary": "Get a specific user by id",
974         "operationId": "getUserByIdUsingGET",
975         "parameters": [
976             {
977                 "name": "id",
978                 "in": "path",
979                 "description": "id",
980                 "required": true,
981                 "style": "simple",
982                 "schema": {
983                     "type": "integer",
984                     "format": "int32"
985                 }
986             }
987         ],
988         "responses": {
989             "200": {
990                 "description": "Success!",
991                 "content": {
992                     "*/*": {
993                         "schema": {
994                             "$ref": "#/components/schemas/UserRes"
995                         }
996                     }
997                 }
998             },
999             "401": {
1000                 "description": "Not authorized!"
1001             },
1002             "403": {
1003                 "description": "Forbidden!"
1004             },
1005             "404": {
1006                 "description": "Not found"
1007             },
1008             "510": {
1009                 "description": "Internal server error"
1010             }
1011         }
1012     },
1013     "put": {
1014         "tags": [
1015             "UpdateUser"
1016         ],
1017         "summary": "Updates an existing user to match the provided user",
1018         "operationId": "updateUserUsingPUT",
1019         "parameters": [
1020             {
1021                 "name": "id",
1022                 "in": "path",
1023                 "description": "id",
1024                 "required": true,
1025                 "style": "simple",
1026                 "schema": {
1027                     "type": "integer",
1028                     "format": "int32"
1029                 }
1030             }
1031         ],
1032         "requestBody": {
1033             "content": {
1034                 "application/json": {
1035                     "schema": {

```

```

1036         "$ref": "#/components/schemas/UserReq"
1037     }
1038 }
1039 }
1040 },
1041 "responses": {
1042     "200": {
1043         "description": "Success!",
1044         "content": {
1045             "*/*": {
1046                 "schema": {
1047                     "$ref": "#/components/schemas/UserRes"
1048                 }
1049             }
1050         }
1051     },
1052     "201": {
1053         "description": "Created"
1054     },
1055     "401": {
1056         "description": "Not authorized!"
1057     },
1058     "403": {
1059         "description": "Forbidden!"
1060     },
1061     "404": {
1062         "description": "Not found"
1063     },
1064     "510": {
1065         "description": "Internal server error"
1066     }
1067 },
1068 },
1069 "delete": {
1070     "tags": [
1071         "DeleteUser"
1072     ],
1073     "summary": "Deletes a user from the database",
1074     "operationId": "deleteUserUsingDELETE",
1075     "parameters": [
1076         {
1077             "name": "id",
1078             "in": "path",
1079             "description": "id",
1080             "required": true,
1081             "style": "simple",
1082             "schema": {
1083                 "type": "integer",
1084                 "format": "int32"
1085             }
1086         }
1087     ],
1088     "responses": {
1089         "200": {
1090             "description": "Success!",
1091             "content": {
1092                 "*/*": {
1093                     "schema": {
1094                         "type": "string"
1095                     }
1096                 }
1097             }
1098         },
1099         "204": {
1100             "description": "No Content"
1101         },
1102         "401": {
1103             "description": "Not authorized!"
1104         },
1105         "403": {
1106             "description": "Forbidden!"
1107         },
1108         "404": {
1109             "description": "Not found"

```

```

1110         },
1111         "510": {
1112             "description": "Internal server error"
1113         }
1114     }
1115 }
1116 },
1117 },
1118 "components": {
1119     "schemas": {
1120         "Iterable": {
1121             "title": "Iterable",
1122             "type": "object"
1123         },
1124         "JsonNode": {
1125             "title": "JsonNode",
1126             "type": "object"
1127         },
1128         "LikedReq": {
1129             "title": "LikedReq",
1130             "type": "object",
1131             "properties": {
1132                 "id": {
1133                     "type": "integer",
1134                     "format": "int32"
1135                 },
1136                 "name": {
1137                     "type": "string"
1138                 },
1139                 "user": {
1140                     "$ref": "#/components/schemas/UserReq"
1141                 }
1142             }
1143         },
1144         "LikedRes": {
1145             "title": "LikedRes",
1146             "type": "object",
1147             "properties": {
1148                 "id": {
1149                     "type": "integer",
1150                     "format": "int32"
1151                 },
1152                 "name": {
1153                     "type": "string"
1154                 }
1155             }
1156         },
1157         "ModelAndView": {
1158             "title": "ModelAndView",
1159             "type": "object",
1160             "properties": {
1161                 "empty": {
1162                     "type": "boolean"
1163                 },
1164                 "model": {
1165                     "type": "object"
1166                 },
1167                 "modelMap": {
1168                     "type": "object",
1169                     "additionalProperties": {
1170                         "type": "object"
1171                     }
1172                 },
1173                 "reference": {
1174                     "type": "boolean"
1175                 },
1176                 "status": {
1177                     "type": "string",
1178                     "enum": [
1179                         "ACCEPTED",
1180                         "ALREADY_REPORTED",
1181                         "BAD_GATEWAY",
1182                         "BAD_REQUEST",
1183                         "BANDWIDTH_LIMIT_EXCEEDED",

```



```

1184         "CHECKPOINT",
1185         "CONFLICT",
1186         "CONTINUE",
1187         "CREATED",
1188         "DESTINATION_LOCKED",
1189         "EXPECTATION_FAILED",
1190         "FAILED_DEPENDENCY",
1191         "FORBIDDEN",
1192         "FOUND",
1193         "GATEWAY_TIMEOUT",
1194         "GONE",
1195         "HTTP_VERSION_NOT_SUPPORTED",
1196         "IM_USED",
1197         "INSUFFICIENT_SPACE_ON_RESOURCE",
1198         "INSUFFICIENT_STORAGE",
1199         "INTERNAL_SERVER_ERROR",
1200         "I_AM_A_TEAPOT",
1201         "LENGTH_REQUIRED",
1202         "LOCKED",
1203         "LOOP_DETECTED",
1204         "METHOD_FAILURE",
1205         "METHOD_NOT_ALLOWED",
1206         "MOVED_PERMANENTLY",
1207         "MOVED_TEMPORARILY",
1208         "MULTIPLE_CHOICES",
1209         "MULTI_STATUS",
1210         "NETWORK_AUTHENTICATION_REQUIRED",
1211         "NON_AUTHORITATIVE_INFORMATION",
1212         "NOT_ACCEPTABLE",
1213         "NOT_EXTENDED",
1214         "NOT_FOUND",
1215         "NOT_IMPLEMENTED",
1216         "NOT_MODIFIED",
1217         "NO_CONTENT",
1218         "OK",
1219         "PARTIAL_CONTENT",
1220         "PAYLOAD_TOO_LARGE",
1221         "PAYMENT_REQUIRED",
1222         "PERMANENT_REDIRECT",
1223         "PRECONDITION_FAILED",
1224         "PRECONDITION_REQUIRED",
1225         "PROCESSING",
1226         "PROXY_AUTHENTICATION_REQUIRED",
1227         "REQUESTED_RANGE_NOT_SATISFIABLE",
1228         "REQUEST_ENTITY_TOO_LARGE",
1229         "REQUEST_HEADER_FIELDS_TOO_LARGE",
1230         "REQUEST_TIMEOUT",
1231         "REQUEST_URI_TOO_LONG",
1232         "RESET_CONTENT",
1233         "SEE_OTHER",
1234         "SERVICE_UNAVAILABLE",
1235         "SWITCHING_PROTOCOLS",
1236         "TEMPORARY_REDIRECT",
1237         "TOO_EARLY",
1238         "TOO_MANY_REQUESTS",
1239         "UNAUTHORIZED",
1240         "UNAVAILABLE_FOR_LEGAL_REASONS",
1241         "UNPROCESSABLE_ENTITY",
1242         "UNSUPPORTED_MEDIA_TYPE",
1243         "UPGRADE_REQUIRED",
1244         "URI_TOO_LONG",
1245         "USE_PROXY",
1246         "VARIANT_ALSO_NEGOTIATES"
1247     ],
1248     },
1249     "view": {
1250         "$ref": "#/components/schemas/View"
1251     },
1252     "viewName": {
1253         "type": "string"
1254     }
1255 }
1256 },
1257 "RestaurantReq": {

```

```

1258     "title": "RestaurantReq",
1259     "type": "object",
1260     "properties": {
1261         "_address": {
1262             "type": "string"
1263         },
1264         "_alias": {
1265             "type": "string"
1266         },
1267         "_code": {
1268             "type": "string"
1269         },
1270         "_count": {
1271             "type": "integer",
1272             "format": "int32"
1273         },
1274         "_distance": {
1275             "type": "string"
1276         },
1277         "_image_url": {
1278             "type": "string"
1279         },
1280         "_is_closed": {
1281             "type": "string"
1282         },
1283         "_latitude": {
1284             "type": "string"
1285         },
1286         "_longitude": {
1287             "type": "string"
1288         },
1289         "_name": {
1290             "type": "string"
1291         },
1292         "_phone_number": {
1293             "type": "string"
1294         },
1295         "_price": {
1296             "type": "string"
1297         },
1298         "_rating": {
1299             "type": "string"
1300         },
1301         "_review_count": {
1302             "type": "string"
1303         },
1304         "_url": {
1305             "type": "string"
1306         }
1307     }
1308 },
1309 "RestaurantRes": {
1310     "title": "RestaurantRes",
1311     "type": "object",
1312     "properties": {
1313         "_address": {
1314             "type": "string"
1315         },
1316         "_alias": {
1317             "type": "string"
1318         },
1319         "_code": {
1320             "type": "string"
1321         },
1322         "_distance": {
1323             "type": "string"
1324         },
1325         "_image_url": {
1326             "type": "string"
1327         },
1328         "_is_closed": {
1329             "type": "string"
1330         },
1331         "_latitude": {

```

```

    "type": "string"
  },
  "_longitude": {
    "type": "string"
  },
  "_name": {
    "type": "string"
  },
  "_phone_number": {
    "type": "string"
  },
  "_price": {
    "type": "string"
  },
  "_rating": {
    "type": "string"
  },
  "_review_count": {
    "type": "string"
  },
  "_url": {
    "type": "string"
  }
},
"UserReq": {
  "title": "UserReq",
  "type": "object",
  "properties": {
    "favoriteRestaurants": {
      "uniqueItems": true,
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/RestaurantReq"
      }
    },
    "halal": {
      "type": "boolean"
    },
    "id": {
      "type": "integer",
      "format": "int32"
    },
    "likes": {
      "uniqueItems": true,
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/LikedReq"
      }
    },
    "name": {
      "type": "string"
    },
    "newLike": {
      "$ref": "#/components/schemas/LikedReq"
    },
    "passkey": {
      "type": "string"
    },
    "username": {
      "type": "string"
    },
    "vegan": {
      "type": "boolean"
    },
    "vegetarian": {
      "type": "boolean"
    }
  }
},
"UserRes": {
  "title": "UserRes",
  "type": "object",
  "properties": {

```

```

1406     "allFriends": {
1407         "uniqueItems": true,
1408         "type": "array",
1409         "items": {
1410             "type": "string"
1411         }
1412     },
1413     "favoriteRestaurants": {
1414         "uniqueItems": true,
1415         "type": "array",
1416         "items": {
1417             "$ref": "#/components/schemas/RestaurantRes"
1418         }
1419     },
1420     "halal": {
1421         "type": "boolean"
1422     },
1423     "id": {
1424         "type": "integer",
1425         "format": "int32"
1426     },
1427     "likes": {
1428         "uniqueItems": true,
1429         "type": "array",
1430         "items": {
1431             "$ref": "#/components/schemas/LikedRes"
1432         }
1433     },
1434     "name": {
1435         "type": "string"
1436     },
1437     "passkey": {
1438         "type": "string"
1439     },
1440     "username": {
1441         "type": "string"
1442     },
1443     "vegan": {
1444         "type": "boolean"
1445     },
1446     "vegetarian": {
1447         "type": "boolean"
1448     }
1449 }
1450 },
1451 "View": {
1452     "title": "View",
1453     "type": "object",
1454     "properties": {
1455         "contentType": {

```