STT	API	REQUEST		METHOD	URL	URL PARAMETERS	RESPONSE DATA	NOTE		ACCOUNT ROOT	
		HEADER	PARAMETERS							email	root@gmail.com
			NAME							pass	root12345

1	LOGIN		{
			email: String,
			userPassword: string
			}
				POST	http://localhost:5000/authenticate		{
							code,
							data : {
							jwtToken,
							email,
							role
							}
							}
								login với account root để tạo tài khoản admin rồi mới dùng đc các API admin			

2	GET_ADMINS	Authorization: token		GET	http://localhost:5000/admins		{
							code,
							data: [
							{
							id,
							username,
							password,
							email,
							role
							},
							{},...
							]
							}
								các api này chỉ có account root dùng đc			

3	GET_ADMIN	Authorization: token		GET	http://localhost:5000/admins/{adminId}	adminId :String	{
							code,
							data: {
							id,
							username,
							password,
							email,
							role
							}
							}				

4	ADD_ADMIN	Authorization: token	{
			username : string,
			password: srting,
			email: string,
			role: admin
			}
				POST	http://localhost:5000/admins/addAdmin		{
							code,
							data: {
							id,
							username,
							password,
							email,
							role
							}
							}
									api này đang trả về pass sau khi mã hóa
									có thể đánh fail vì mình muốn origin pass chẳng hạn		

5	EDIT_ADMIN	Authorization: token	{
			username : string,
			password: srting,
			email: string,
			role: admin
			}
				PUT	http://localhost:5000/admins/editAdmin/{adminId}	adminId :String	{
							code,
							data: {
							id,
							username,
							password,
							email,
							role
							}
							}				

6	DELETE_ADMIN	Authorization: token		DELETE	http://localhost:5000/admins/deleteAdmin/{adminId}	adminId :String	{
							code,
							data: "Delete admin has id ..."
							}				

7	GET_USERS	Authorization: token		GET	http://localhost:5000/users/{page}/{size}?searchBy={field1}&searchValue=&sortBy={field2}&sortValue	page: int, size: int,
						searchBy: string(1 field của object user),
						searchValue: String (giá trị search)(not require),
						sortBy: string((1 field của object user),
						sortValue: "ASC" hoặc "DESC"(not require)
							{
							code,
							data: [
							{
							id,
							name,
							email,
							newestBorrowDate,
							borrowTime
							},
							{},...
							]
							}				

8	GET_USER	Authorization: token		GET	http://localhost:5000/users/{userId}	userId: string	{
							code,
							data: {
							id,
							name,
							email,
							newestBorrowDate,
							borrowTime
							},
							}				

9	ADD_USER	Authorization: token	{
			name: String,
			email: string,
			newestBorrowDate: date,
			borrowTime: int
			}
				POST	http://localhost:5000/users/addUser		{
							code,
							data: {
							id,
							name,
							email,
							newestBorrowDate,
							borrowTime
							}
							}
								các api này chỉ có account admin dùng đc			

10	EDIT_USER	Authorization: token	{
			"name" : string,
			"email" : String,
			"newestBorrowDate": date
			"borrowTime" : int
			}
				PUT	http://localhost:5000/users/editUser/{userId}	userId: string	{
							code,
							data: {
							id,
							name,
							email,
							newestBorrowDate,
							borrowTime
							}
							}				

11	DELETE_USER	Authorization: token		DELETE	http://localhost:5000/users/deleteUser/{userId}	userId: string	{
							code,
							data: "Delete user has id ..."
							}				

12	GET_BOOKS	Authorization: token		GET	http://localhost:5000/books/{page}/{size}?searchBy={field1} &searchValue=&sortBy={field2}&sortValue	page: int, size: int, searchBy: string(1 field của object user), searchValue: String (giá trị search)(not require), sortBy: string((1 field của object user), sortValue: "ASC" hoặc "DESC"(not require)	"code": 0, "data": [{ "id": String, "name": String, "author": String, "publishDate": Date, "price": int }, {},...]				

13	GET_BOOK	Authorization: token		GET	http://localhost:5000/books/{bookId}	bookId :string	"code": 0,
							"data": {
							"id": String,
							"name": String,
							"author": String,
							"publishDate": Date,
							"price": int
							}				

14	ADD_BOOK	Authorization: token	{
			"name" : string,
			"author" : string,
			"publishDate" : Date,
			"price" : int
			}
				POST	http://localhost:5000/books/addBook		"code": 0,
							"data": {
							"id": String,
							"name": String,
							"author": String,
							"publishDate": Date,
							"price": int
							}				

15	EDIT_BOOK	Authorization: token	{
			"name" : string,
			"author" : string,
			"publishDate" : Date,
			"price" : int
			}
				PUT	http://localhost:5000/books/editBook/{bookId}	bookId :string	"code": 0,
							"data": {
							"id": String,
							"name": String,
							"author": String,
							"publishDate": Date,
							"price": int
							}				

16	DELETE_BOOK	Authorization: token		DELETE	http://localhost:5000/deleteBook/{bookId}	bookId :string	{
							code,
							data: "Delete book has id ..."
							}				

17	GET_CARD_BORROWS	Authorization: token		GET	http://localhost:5000/borrowed-information		{
							"code": 0,
							"data": [
							{
							"id": String,
							"bookId": String,
							"userId": String,
							"adminId": String,
							"borrowDate": date,
							"borrowTime": int
							}, {},....
							]
							}				

18	GET_CARD_BORROW	Authorization: token		GET	http://localhost:5000/borrowed-information/{borrowCardId}	borrowCardId: string	"code": 0,
							"data": {
							"id": String,
							"bookId": String,
							"userId": String,
							"adminId": String,
							"borrowDate": date,
							"borrowTime": int
							}				

19	ADD_CARD_BORROW	Authorization: token	{
			bookId: string,
			userId: string,
			adminId: String,
			borrowTime: int
			}
				POST	http://localhost:5000/borrowed-information/addBorrowCard		"code": 0,
							"data": {
							"id": String,
							"bookId": String,
							"userId": String,
							"adminId": String,
							"borrowDate": date,
							"borrowTime": int
							}				

20	EDIT_CARD_BORROW	Authorization: token	{
			bookId: string,
			userId: string,
			adminId: String,
			borrowTime: int
			}
				PUT	http://localhost:5000/borrowed-information/editBorrowCard/{borrowCardId}	borrowCardId: string	"code": 0,
							"data": {
							"id": String,
							"bookId": String,
							"userId": String,
							"adminId": String,
							"borrowDate": date,
							"borrowTime": int
							}				

21	DELETE_CARD_BORROW	Authorization: token		DELETE	http://localhost:5000/borrowed-information/deleteBorrowCard/{borrowCardId}	borrowCardId: string	{
							code,
							data: "Delete borrow card has id ..."
							}				