



Thinh Nguyen Viet

Junior Javascript Web Developer

[My Online](#)



A hard-working developer always wants to create optimal solutions that satisfy both teammates and customers.

Passionate about learning new languages and technologies.

Concern about writing clean, readable, best practices code.

Good English communication skills.

Seeking to become the next **NodeJS** back-end developer at **your company**.



Experience

```
const thinhnv = {  
  code: [ 'JS', 'HTML', 'CSS' ],  
  technologies: {  
    backEnd: [ 'Node', 'Express' ],  
    frontEnd: [ 'Vue', 'Vuex', 'Vue Router',  
    'Vuetify' ],  
    database: [ 'MongoDB', 'Mongoose' ]  
  },  
  tools: [ 'Git', 'GitHub', 'VsCode',  
  'ESLint', 'Prettier' ],  
  futureGoal: [ 'React', 'Microservices',  
  'Design Pattern', 'Docker', 'Jest',  
  'Typescript', 'SQL' ],  
  wannaTry: [ 'Deno', 'Svelte' ],  
};
```

Thinh Nguyen's GitHub Stats

	Total Commits:	248
	Total PRs:	2
	Total Issues:	11
	Contributed to:	8



• 2014:

Graduated from [Pune University \(India\)](#).

• 2014-2019:

Got several jobs as webmaster/IT technician/designer in some small MMO teams.

• Late 2019-2020:

Quit the job and followed the web developer path.

Self-taught **JS, Node.js, Vue.js** in several months.



Education



Pune University

(Pune city – India)

2011 - 2014

Bachelor in Computer Applications



Hanoi University of Science and Technology

2010 – 2011

Information Technology



Personal Information



Nam Tu Liem, Ha Noi



maxiqboy@gmail.com



[Github.com/thinh105](https://github.com/thinh105)



091 682 7859



[Messenger.com/
t/Thinh105](https://messenger.com/t/Thinh105)



[Stackoverflow.com/users/
2620022/thinh-nv](https://stackoverflow.com/users/2620022/thinh-nv)



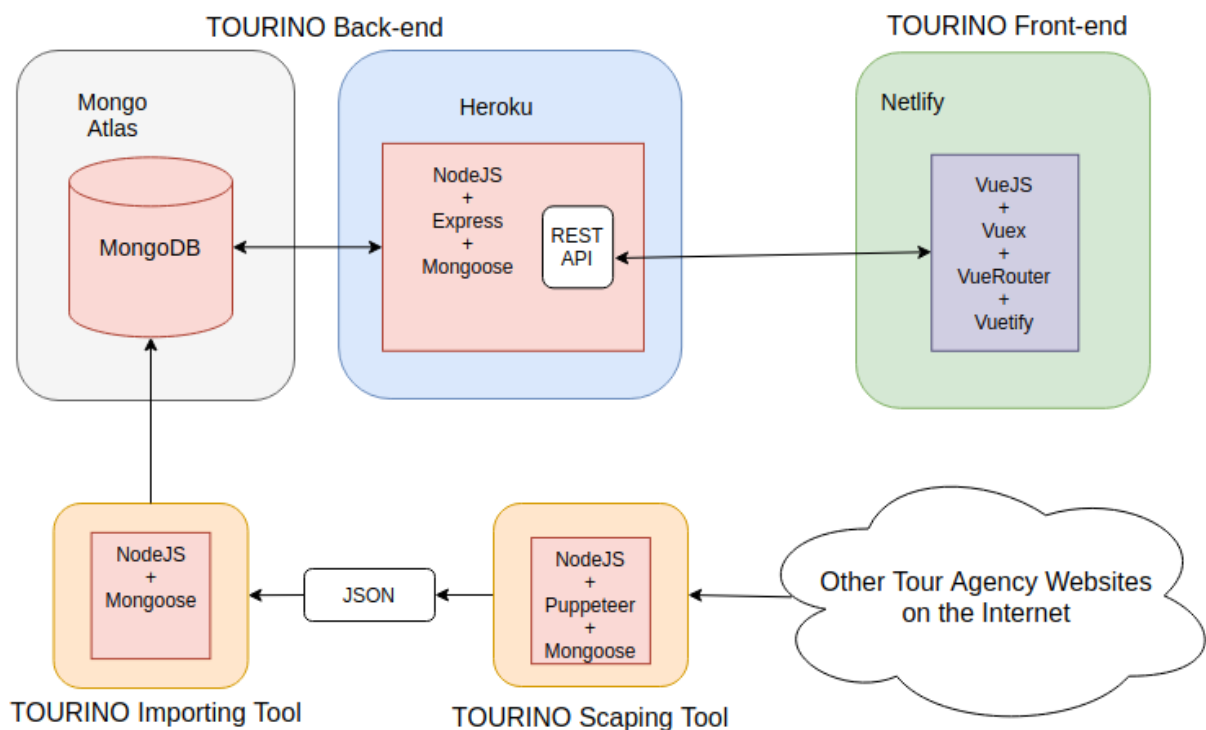
TOURINO, my personal project, is a fullstack Travel Agency Website powered by **NodeJS**, **ExpressJS**, **MongoDB**, **VueJS**.

Live Demo: <https://tourino.netlify.app/>

Notice: It will take a while (around 10-15s) to wait and fetch data when you visit it first time.

Because I run the **TOURINO** 's backend under the free package on Heroku, my API server will sleep after 30m of inactivity.

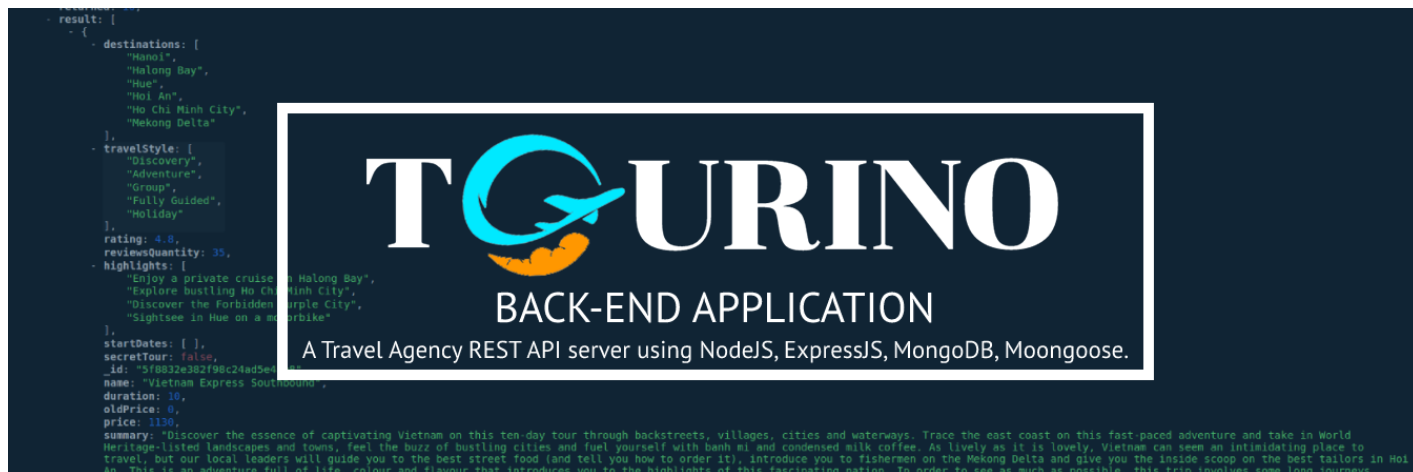
Application Architecture Diagram



Code Base

It contains three parts (**TOURINO** Importing Tool is inside the **TOURINO** Backend Nodejs):

- 1: Tourino-Backend-Nodejs: <https://github.com/thinh105/Tourino-Backend-Nodejs>
- 2: Tourino-Frontend-VueJS: <https://github.com/thinh105/Tourino-Frontend-VueJS>
- 3: Tourino-Scraping-Nodejs: <https://github.com/thinh105/Tourino-Scraper-Nodejs>



About this project

A REST API Backend using **NodeJS - ExpressJS - Mongoose** for the **TCURINO** Project contains practical examples (CRUD, Authentication & Authorization, Error Handling, Security).



More info
in **GitHub**



API Spec

Live Demo : <https://tourino.herokuapp.com/api/v1/>



Tours

	HTTP Verb	POST Body	Authentication	Authorization	Result
/api/v1/tours	GET				Return all tours
/api/v1/tours/:id	GET				Return single tour (based on id)
/api/v1/tours/slug/:slug	GET				Return single tour (based on slug)
/api/v1/tours	POST	JSON String	Yes	moderator, admin	Create new tour
/api/v1/tours/:id	PATCH	JSON String	Yes	moderator, admin	Updates an existing tour
/api/v1/tours/:id	DELETE		Yes	moderator, admin	Deletes an existing tour
-					
/api/v1/tours/destinations	GET				Return all destinations and count
/api/v1/tours/travelStyle	GET				Return all travel style and count



Users


URL	HTTP Verb	POST Body	Authentication	Authorization	Result
/api/v1/users/signup	POST	JSON String			Create a new user
/api/v1/users/login	POST	JSON String			Create a new user
/api/v1/users/forgotPassword	POST	JSON String			Create a new user
/api/v1/users/resetPassword:token	POST	JSON String			Create a new user
-					
/api/v1/users/me	GET		Yes		Return a current user
/api/v1/users/updateMyPassword	PATCH	JSON String	Yes		Updates password for a current user
/api/v1/users/updateMe	PATCH	JSON String	Yes		Updates a current user
/api/v1/users/deleteMe	DELETE		Yes		Deletes a current user
-					
/api/v1/users	GET		Yes	admin	Return all users
/api/v1/users/role	GET		Yes	admin	Return all Role's types and count
/api/v1/users/:id	GET		Yes	admin	Return a specific user
/api/v1/users/:id	PATCH	JSON String	Yes	admin	Update a specific user
/api/v1/users/:id	DELETE		Yes	admin	Delete a specific user



Reviews

URL	HTTP Verb	POST Body	Authentication	Authorization	Result
/api/v1/tours/:tourId/reviews	GET				Return all reviews from a specific tour
/api/v1/reviews	GET				Return all reviews
/api/v1/reviews/:id	GET				Return a specific review
/api/v1/reviews	POST	JSON String	Yes	user	Create new review
/api/v1/reviews/:id	PATCH	JSON String	Yes	user(owner of review), admin	Updates a specific review
/api/v1/reviews/:id	DELETE		Yes	user(owner of review), admin	Deletes a specific review



 **About this project:** A Single Page Application Front-end using **Vue.js, Vuex, Vuetify, Axios** for the **TOURINO** Project.



 **Live Demo:** <https://tourino.netlify.app/>

Notice: It will take a while (around 10-15s) to wait and fetch data when you visit my website the first time. Because I run the **TOURINO**'s backend under the free package on Heroku, my API server will sleep after 30m of inactivity.

I want to play fair and will not use some tricks to abuse their server every 29 minutes just to keep my small project from sleeping. It's kind of a feature, not a bug! 😊 😊

Features

- Unique Responsive UI that was designed and coded from scratch, not using any pre-made template on the internet.
- Fetch and show tours with paginations from Backend API.
- Filter tours by
 - Destinations (multi-choice)
 - Travel Style (multi-choice)
 - Price (range)
 - Rating
 - Duration
- Sort tours by
 - Price (ascending and descending order)
 - Duration (ascending and descending order)
 - Rating
 - Reviews
 - New on Tourino
- Sync the Sort and filter selections with URL
 - User can copy the URL and send to people, and they can visit that link to get the same result as User
 - User can click the back/forward button on the browser and the selections will be sync with the URL.
- Fetch and show detail tour page from Backend API.
- Authentication & Authorization
 - Log in, sign up, reset the password.
 - If an anonymous visit the authorized link, they will be redirected to the Authentication page.
- Error handling
 - When Backend API are not response, display Error messages from API.

Account Demo:

Email: alicebui@mailinator.com | nicolehuynh@mailinator.com | mariedang@mailinator.com

Pass: **test1234**