# Api 设计

## api.eesast.com/v1

### 静态资源

上传文件名增加 uuid.

api.eesast.com/static/

/images/example.png

/files/example.docx

#### GET

/statics/:category/:filename

@returns static file

#### POST

/statics/:category/

@param {String} category 文件夹

@returns {String} Location header

#### DELETE

/statics/:category/:filename

@param {String} category 文件夹

@param {String} filename 文件名

@returns {String} No Content 或 Not Found

### 文章

/articles

{

id: Number, // use auto-increment id, instead of \_id generated by database

title: String, // “TensorFlow 初探”

alias: String, // for SEO and URL, “tensor-flow-first-look”

authorId: Number,

abstract: String, // 摘要，首页用

image: String, // 题图，首页用，URL

content: String, // markdown

views: Number, // 阅读量

likes: Number, // 点赞量

likers: [Number], // 点赞人 id

tags: [String],

createdAt: { type: Date, default: Date.now },

createdBy: Number,

updatedAt: { type: Date, default: Date.now },

updatedBy: Number

}

#### GET

/articles

@returns all articles

/articles/:id

@param {Number} id

@returns article with id

/articles?

@query {String} title

@query {String} author

@query {String} tag

@query {Number} likedBy

@returns certain articles

#### POST

/articles

@returns {String} Location header

#### PUT

/articles/:id

@param {Number} id 需要更新的文章 id

@returns {String} Location header 或空

#### DELETE

/articles/:id

@param {Number} id 需要删除的文章 id

@returns {String} No Content 或 Not Found

### 用户

权限系统采用 RBAC

/users/:id

{

id: Number, // 用户的学工号

username: String,

group: String, // 用户组：admin - 管理员，user - 普通用户  
role: [String], // 分配用户角色，便于权限管理。  
password: String,  
email: String,

name: String,  
phone: Number,  
department: String,  
class: String,  
createdAt: { type: Date, default: Date.now },

createdBy: Number,

updatedAt: { type: Date, default: Date.now },

updatedBy: Number

}

#### GET

/users

@returns all users

/users/:id

@param {Number} id

@returns user with id

/users?

@query {String} username

@query {String} group

@query {String} role

@query {String} email

@query {String} name

@query {Number} phone

@query {String} department

@query {String} class

@returns certain users

POST

/users

@returns {String} Location header

/users/login

@returns {String} token

@returns {Boolean} auth

// 找回密码，发送邮件，包含 /users/forgot/:encoded 链接

/users/forgot

@returns {String} No Content 或 Not Found

#### PUT

/users/:id

// 除了用户权限控制外，可以额外传入一个编码参数，如果该编码解码后信息与用户信息一致，且未过期，则可以直接修改密码。

@param {Number} id 需要更新的用户 id

@returns {String} Location header 或空

#### DELETE

/users/:id

@param {Number} id 需要删除的用户 id

@returns {String} No Content 或 Not Found

### 评论

/comments/:id

{

id: Number, // use auto-increment id, instead of \_id generated by database

articleId: Number, // 评论文章的 id

content: String, // markdown

replyTo: Number, // null 表示是顶层回复，非 null 表示回复给某一 id 的 comment

likes: Number, // 点赞量

likers: [Number], // 点赞人 id

createdAt: { type: Date, default: Date.now },

createdBy: Number,

updatedAt: { type: Date, default: Date.now },

updatedBy: Number

}

#### GET

/comments

@returns all comments

/comments/:id

@param {Number} id

@returns comment with id

/comments?

@query {Number} replyTo

@query {Number} likedBy

@query {Number} articleId

@query {Number} authorId

@returns certain articles

#### POST

/comments

@returns {String} Location header

#### PUT

/comments/:id

@param {Number} id 需要更新的评论 id

@returns {String} Location header 或空

#### DELETE

/comments/:id

@param {Number} id 需要删除的评论 id

@returns {String} No Content 或 Not Found

### 物品

/items/:id

{

id: Number, // auto-increment id

name: String,

description: String,

total: Number,

left: Number,

createdAt: { type: Date, default: Date.now },

createdBy: Number,

updatedAt: { type: Date, default: Date.now },

updatedBy: Number

}

#### GET

/items

@returns all items

/items/:id

@param {Number} id

@returns item with id

/items?

@query {String} name

@query {Number} left

@returns certain items

#### POST

/items

@returns {String} Location header

#### PUT

/items/:id

@param {Number} id 需要更新的物品 id

@returns {String} Location header 或空

#### DELETE

/items/:id

@param {Number} id 需要删除的物品 id

@returns {String} No Content 或 Not Found

### 预约

/reservations/:id

{

id: Number, // auto-increment id

//

itemId: Number, // 预约物品的 id

userId: Number, //预约者 id

from: Date,

to: Date,

createdAt: { type: Date, default: Date.now },

createdBy: Number,

updatedAt: { type: Date, default: Date.now },

updatedBy: Number

}

#### GET

/reservations

@returns all reservations

/reservations/:id

@param {Number} id

@returns reservation with id

/reservations?

@query {Number} itemId

@query {Number} userId

@query {Date} from

@query {Date} to

@returns certain reservations

#### POST

/reservations

@returns {String} Location header

#### PUT

/reservations/:id

@param {Number} id 需要更新的预约 id

@returns {String} Location header 或空

#### DELETE

/reservations/:id

@param {Number} id 需要删除的预约 id

@returns {String} No Content 或 Not Found