2022 퍼블리셔 부트캠프 017

- 예제: Newsletter sign-up (MailChimp) / express.static() / https.request()

!Tips & Links

※ 강의 https://www.udemy.com/course/the-complete-web-development-bootcamp/

※ MailChimp https://mailchimp.com/ Dev https://mailchimp.com/developer/

      DOC https://mailchimp.com/developer/marketing/docs/fundamentals/#connecting-to-the-api

※ https (nodejs 기본module): https://nodejs.org/api/https.html

※ 부트스트랩 템플릿 챙길 거: ① html, ② CDN, ③ 템플렛별 CSS<!-- Custom styles for this template -->

※ 세트처럼 쓰임: Exress, BodyParser Install / HTTPS는 내장 ※ Form에는 action, method

---

Localhost (Sever) & External links (Static files)	①	href에 full path:	"http://localhost:3000/css/~~.css"
	②	**express.static**	app.use(express.statc(' dirname '))

MailChimp : 메일레터 API (회원정보 수집, 캠페인, 메일링) // GET&POST json으로 주고 받음

※ Key, ID 발급

    API key = APP Key
    Audience ID = List ID

API Key	account> extras> API Key			
List ID	audience> all contacts> settings> audience name and campaign default> Audience ID			
URL 적용	end point	https://usX.api.mailchimp.com/3.0/	optional path	lists/{list_id}
		usX는 API Key 마지막 us1~20		= audience ID
auth 적용	options부분 object에 { auth: "string(any name) : API Key" }			

※ field설정! merge_fields 변경하기 => 홈페이지 로그인,

    all audience> audience fields and MERGE tags에서 필드 수정, default 권장 :-)

---

https.request() : https.request(url=string, options=object, callback function), returns http.ClientRequest

```
app.post("/", function(req, res){                    const url='https://us~.api.mailchimp.com/3.0/lists/~'
  const name=req.body.name;                          const options= {              // options 포장
  const nickname=req.body.nickname;                    method: 'POST',
  const email=req.body.email;                          auth: "~~API KEY~~"
  const data={               // object data          }
    members: [{                                      const request=https.request(url, options, function(response){
       email_address: email,                           response.on('data', function(data){ //post에 data res가 있으면
       status: "subscribed",                              console.log(JSON.parse(data));   //data를 JSON으로(parse)
       merge_field: {         // field로 보냄            });                               object로 console.log
         FNAME: name,                                  });
         NNAME: nickname                               request.write(jsonData);          // jsonData를 request하고
    }}]};                                              request.end();                    // 끝
  const jsonData=JSON.stringify(data); // jsondata  });
```

```
/* -----<lang: en>--------------------------------------------------------------------------------- */
to serve up static files, such as CSS, images ☞ use express.static / builit-in middleware
remote( server ) <-> local( static ) // auth ✎ Authentication[ôˌTHen(t)iˈkāSH(ə)n]
https url = endpoint + optional path
```