

캡스톤 디자인(2) 7 조 최종 보고서



과 목 : 캡스톤 디자인(2)

제 출 일 자 : 2021. 06. 08

담 당 교 수 : 박상오 교수님

학 과 : 컴퓨터공학부,
소프트웨어공학부

팀 원 : 20155951 김준오
20170457 이예주
20150594 이종아

목차

1. 소개

1) 프로젝트 이름	03
2) 팀 소개	03
3) 프로젝트 소개	03
4) 제안 동기	04
5) 타겟층 및 개발목표	05
6) 차별성	05
7) 팀원 역할	06

2. 프로젝트 세부 사항

1) Front-end.....	07
a. 개발 환경	07
b. 개발 기능.....	07
2) Back-end.....	11
a. 개발 환경	23
b. 구현 사항	23
c. API 명세서	26

3. 결론

1) Github repository	67
2) 기대효과.....	67
3) 아쉬웠던 점	67

소개

프로젝트 이름

Anipal

팀 소개

- 컴퓨터 공학부 20155951 김준오
- 컴퓨터 공학부 20150594 이종아
- 컴퓨터 공학부 20170457 이예주

프로젝트 소개

사용자가 하고 싶은 말을 담아 편지를 작성하게 되면, 사용자가 보유한 귀여운 동물이 편지를 들고 답장을 해줄 누군가를 찾아 세계 각국을 떠돌아다니며 답장을 받아와 다양한 사람들과 소통을 하게 해주는 펜팔 앱입니다.

해당 동물은 흥미롭게 대화를 나눌 수 있을 것으로 예상되는 사람들을 찾아주는 알고리즘에 의해 선택된 유저의 집을 찾아 돌아다니게 되며, 편지를 확인한 누군가가 해당 편지에 답장을 원할 경우 두 유저 간의 대화가 시작됩니다.

각 사용자는 자신이 보유한 동물들을 통해 편지를 전달할 수 있으며, 동물들의 고유 속도에 따라서 편지의 배송시간이 달라집니다. 이를 통해 유저의 취향대로 편지의 전송시간을 조절할 수 있으며, 편지를 배달하는 동물 역시 사용자의 취향대로 꾸밀 수 있습니다.

편지를 전달 중인 동물은 배송이 끝나고 집으로 돌아올 때까지 다시 사용할 수 없기에 보유하고 있는 동물의 수만큼만 동시에 편지를 전송할 수 있습니다. 이 때 사용하는 동물들을 다양한 미션을 통해서 획득할 수 있습니다.

여러 미션을 통해서 동물들을 꾸밀 옷과 액세서리를 획득하여 사용자의 취향대로 동물을 꾸며주고, 예쁘게 꾸민 동물을 통해 다른 사용자들에게 자신에게 답장해줄 것을 어필할 수 있습니다.

제안 동기

코로나 시대로 맞이하게 된 Untact 환경

2020 년 코로나의 등장과 함께 지금껏 경험해 본 적 없는 새로운 국면의 사회가 만들어졌습니다.

대학교 강의를 포함한 모든 사회 환경은 비 대면, 비 접촉 사회로 접어들었고, 재택근무, 화상회의, 전자 결제, 비 대면 수업 등의 언택트(Untact)가 자리 잡은 지도 어느새 1 여년이 지나고 있습니다. 외출 및 모임 자제로 인한 사회적 고립이 장기적으로 지속되며 국민의 40%이상이 우울감을 호소하고 있는 사회 속, 포스트 코로나 시대에 걸 맞는 온택트(Ontact) 방식의 소통을 생각해보게 되었습니다.

느림의 미학

모든 것이 ‘빨리빨리’가 되어버린 현대화 사회속에서 여유와 아날로그 감성을 느낄 수 있는 느린 편지라는 키워드를 생각해보게 되었습니다. 카카오톡, 인스타그램, 페이스북 과 같은 실시간 소통방식이 아닌, 사용자의 요구에 맞춰 소통의 속도를 조절할 수 있는 소통의 창구가 존재한다면 어떨까 하는 생각해서 기획해보게 되었습니다.

동물의 숲

2020 년 히트를 쳤던 동물의 숲에서 영감을 얻어 다양한 미션 등을 통해 동물들을 수집하고, 각각의 동물들을 내 취향대로 커스터마이징 하여 꾸밀 수 있는, 소유욕을 불러일으킬 수 있는 즐길 거리를 생각해보게 되었습니다. 세계 어느 곳에서나 국적과 문화에 상관없이 동물은 존재하기에 글로벌 시장을 타겟팅 하기에도 동물이라는 컨셉은 적합할 것이라 생각하였습니다.

타겟층 및 개발 목표

타겟층

느린 편지, 느린 필름 등 빨리빨리 가 아닌 천천히 흘러가는 아날로그 감성을 좋아하는 유저들을 주 타겟층으로 삼았습니다. 현대화된 사회 속에서 옛날 감성으로 돌아가 편지를 주고받으며 기다리는 설렘을 얻고자 하는 사람들, 외출 및 모임 자제로 새로운 사람들과 소통할 기회를 잃어버린 사람들, 특정한 목표없이 자신만의 동물을 꾸미고 자랑하고 싶은 사람들, 편하게 고민을 털어놓고 하루를 마무리할 수 있는 여유로운 즐거움을 원하는 사람들을 주 타겟층으로 삼았습니다.

개발 목표

외출 및 모임 자제로 새로운 사람을 만날 기회를 잃어버린 세계 각국의 사람들에게 다양한 문화를 가진 새로운 사람들과 소통할 수 있는 기회를 제공하여 코로나블루 및 우울감으로 힘들어 하는 사람들에게 새로운 즐길 거리를 선사하고 싶습니다.

바쁜 현실 속에서 벗어나 펜팔을 하는 동안만큼은 여유를 느끼며 편하게 고민을 털어놓고 하루를 마무리할 수 있는, 특정한 목표없이 자유롭게 자신만의 동물을 꾸미고 자랑하며 소소한 재미를 느낄 수 있는 여유 있는 즐거움을 선사하고 싶습니다.

차별성

동물의 수집과 커스터마이징

가장 큰 차별점은 동물이라는 컨셉의 활용도라고 생각합니다.

펜팔이라는 요소를 가지고 시작했지만 어느 순간부터는 동물들을 수집하고 키우는 재미, 예쁜 동물들을 사람들에게 자랑하는 재미로도 어플을 사용하게끔 만드는 것이 기획 방향이며 차별성입니다. 기존의 어떤 펜팔 어플도 동물이라는 컨셉이 접목된 것은 없습니다.

펜팔이라는 기본기능도 충실히 제공해주지만, 자연스러운 어플의 사용 속에서 나만의 동물들을 수집하고, 꾸미고 다른 사용자들에게 자랑할 수 있는 게임적인 요소를 녹여낸 것이 어플의 가장 큰 차별점이라고 생각합니다

원하는 대로 조절할 수 있는 전송 시간

유사 어플들과의 또 다른 차별점으로는 편지의 배송 시간을 편의에 맞게 유동적으로 조절할 수 있다는 것입니다. 'Slowly'라는 어플은 이름에서 알 수 있듯이 느린 편지라는 컨셉을 가진 어플입니다. 사용자 간의 실질적인 거리를 바탕으로 편지 전송시간이 정해집니다. 이는 사용자 간 서로 편지를 주고받는 전달 시간이 자동으로 지정되기에 사용자의 요구가 반영되지는 못합니다.

'Bottled'라는 애플리케이션은 '편지를 작성해서 병에 담아 보내면 (무작위의) 누군가가 답장을 해준다'는 랜덤요소가 있다는 점에서 저희와 유사한 면이 있습니다. 하지만 한번 유저와 연결되고 나면 'Bottled'애플리케이션은 실시간 채팅으로 전환되기 때문에 전송시간을 조절할 수 없습니다.

하지만 저희가 만들고자 하는 어플은 동물의 선택을 통해 편지를 주고받고 싶은 대화의 템을 사용자가 스스로 조절할 수 있습니다. 기본적으로 느린 편지라는 방향성을 가지고 있지만 말이 잘 통하고 친해진 상대와는 덜 느리게 주고받을 수도 있으며 아직 친밀감이 쌓이지 않은 상대와는 아주 느리게 주고받을 수도 있습니다. 또한 할 일이 많아 바쁜 시기에는 느리게 전송을 하다가 심심한 시기에는 비교적 빠르게 답을 보내는 등 사용자가 스스로 커스텀 할 수 있는 편지의 전송 시간이 기존 어플들과는 다른 차별점이라고 생각합니다.

HelloTalk, Tandem 의 경우는 언어 학습을 목적으로 하는 앱으로 전반적인 컨셉과 방향이 다릅니다. 해당 앱들은 실제 본인의 사진을 프로필로 등록하고 언어 자체를 학습하는 분위기 이기에 익명으로 하고 폰 이야기나 고민거리들을 편하게 누군가에게 흘려보내고 일과 후 남는 시간에 소통을 한다는 우리 어플과는 방향성이 다릅니다.

팀원 역할

- 김준오 - Client Application (IOS)
- 이예주 - Client Application (IOS), Design
- 이종아 - 팀장 / Server 및 Database 구축, API 설계

프로젝트 세부 사항

Front-end

개발 환경

- IOS: Xcode (version 12.5)
- Simulator : iPhone 12 IOS 14.5

개발 기능

1. 회원가입 및 로그인, 자동 로그인

첫째로 회원가입, 로그인, 자동로그인 기능입니다. 소셜 계정을 통해서 서비스를 이용할 수 있으며, 현재 Google 계정 및 Facebook 계정으로 이용이 가능합니다.

사용자의 프로세스를 간단히 하기 위해 회원가입과 로그인 버튼을 통합했으며, 기존에 가입된 적이 없는 신규 회원일 경우 회원가입 프로세스로 넘어가고, 가입되어있는 회원일 경우 메인 화면으로 넘어갑니다.

만약 회원가입을 시도했지만, 동일 이메일을 사용하는 다른 소셜 계정이 이미 등록되어 있는 경우, 사용자의 실수로 인한 중복 가입을 방지하기 위해 동일 이메일로 가입된 계정이 있음을 알리는 화면을 출력합니다.

한 기기에서 한번 로그인을 진행하였다면, 사용자가 로그아웃을 하기 전까지 자동로그인이 지원됩니다. 앱을 강제로 종료하여도 재 실행 시 별도의 로그인 절차 없이 메인 화면으로 바로 이동합니다.

회원가입시 이름 생년월일, 성별, 관심사, 사용가능언어, 대표동물 선택 등을 입력 받고, 이름과 생년월일을 제외한 데이터는 추후 환경설정 탭에서 수정이 가능합니다.

2. 사용가능 언어 및 관심사

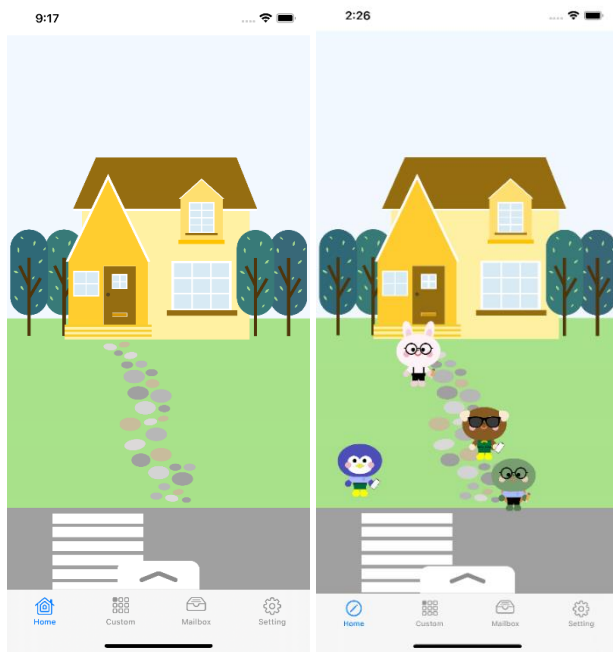


사용가능 언어의 경우 언어 종류 + 레벨을 입력하게 되며, 사용 가능한 언어가 여러가지 일 경우 중복 입력이 가능합니다. 추후 편지를 주고받게 될 경우 내가 어떤 언어에 대해서 어느 정도의 능력을 가지고 있는지 대화의 상대방에게 제공됩니다. 현재 애플에서 지원하는 언어 중 상위 40 개 언어를 채택했습니다.

관심사의 경우 영화, 여행, 노래, 피트니스, 디자인, 요리, 춤, 독서, 미용, 쇼핑, 게임, IT, 건축, 경제 총 12 개의 관심분야를 지원하고 있고, 역시 상대방에게 나의 정보를 제공하는 용도로 사용됩니다. 초기 관심분야는 위 12 가지 분야로 설정하였지만, 점차 확장해 나갈 예정입니다.

입력 받은 언어와 관심사 정보는 또한 랜덤 편지 작성 시 매칭 알고리즘으로도 활용되며, 기본적으로 1 가지 이상의 언어가 겹치는 상대방에게 편지가 전달되고, 그 중에서 관심사 정보가 유사한 상대방이 우선적으로 매칭됩니다.

3. 메인 화면



어플리케이션에 로그인하면 다음과 같은 화면이 나옵니다. 이 화면이 저희 어플리케이션의 메인 화면입니다. 하단 탭바에는 Home, Custom, Mailbox, Setting 이 있습니다.

다른 유저들로부터 도착한 랜덤 편지들을 확인할 수 있습니다. 도착한 편지가 없을 경우 왼쪽 사진처럼 비어 있게 되고, 편지가 왔을 경우 오른쪽 사진처럼 편지를 들고 온 동물이 표시됩니다. 동물의 모습은 편지를 보낸 사람이 꾸며서 보낸 대로 표시되며, 동물의 위치는 동적으로 이동합니다.

3-1. 편지 확인 및 삭제



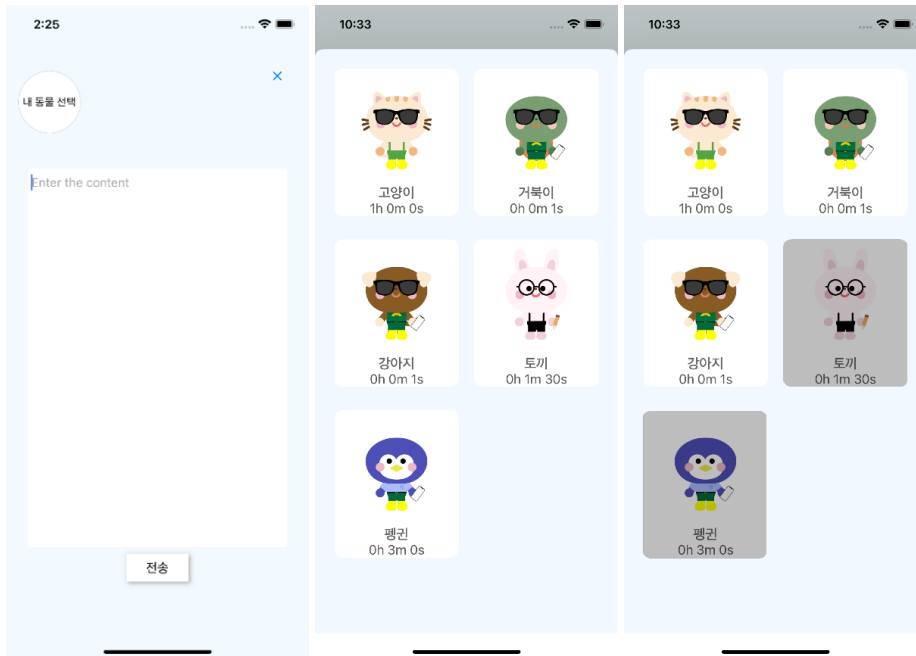
각 동물은 버튼으로 구성되어 있으며, 클릭 시 동물이 들고 온 편지의 내용을 확인할 수 있습니다. UI 는 위와 같이 구성되어 있으며, 상대방의 닉네임과 관심사, 사용 가능한 언어 정보 등을 확인할 수 있고, 편지를 보낸 날짜와 내용을 확인할 수 있습니다.

편지의 내용을 확인하고 불량하거나 마음에 들지 않는 편지일 경우 삭제버튼을 눌러 더이상 내 메인 화면에 표출되지 않게 할 수 있으며, 차단 버튼을 눌러 해당 사용자를 차단 목록에 추가할 수도 있습니다. 사용자 차단 시 해당 사용자가 보내오는 편지는 더이상 나에게 노출되지 않게 됩니다.

좌측 상단의 X 버튼을 누르면 편지 창이 닫히지만 메인 화면에서 사라지지 않습니다. 닫아 두고 나중에 다시 확인이 가능하지만, 다른 사용자가 먼저 답장을 할 경우 사라질 수 있으니 주의해야합니다.

답장 버튼을 눌러 해당 사용자와 소통을 이어갈 수도 있습니다.

3-2. 편지 답장



답장하기 버튼을 눌러 해당 유저에게 답장을 보낼 수 있습니다.

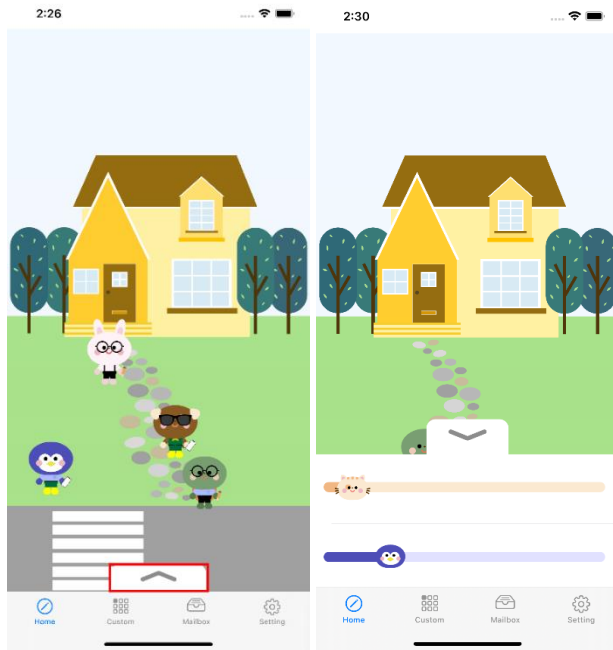
답장을 보낼 경우 해당 동물은 편지를 들고 돌아가게 되며, 이미 매칭이 진행되었기 때문에, 다른 사용자들에게 더이상 노출되지 않게 됩니다.

편지를 보낼 때는 내 동물 선택 버튼을 눌러 배송을 해줄 동물을 선택하게 되며, 동물 별로 편지 배송에 소요되는 배송 시간이 다릅니다.

한번 배송을 떠난 동물은 배송을 마치고 돌아올 때 까지는 선택할 수 없기에, 가지고 있는 동물이 개수만큼 편지를 동시에 보낼 수 있습니다.

동물 리스트는 내가 꾸며 둔 모습대로 표출되며, 상대방에게 역시 해당 모습으로 표시됩니다.

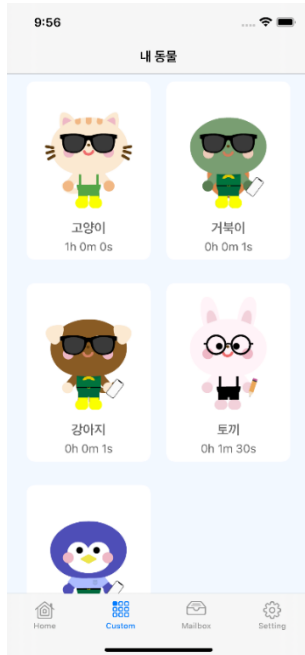
3-3. 도착 예정 편지 확인



하단의 화살표버튼을 눌러서 나에게 오고 있는 편지들을 확인할 수 있습니다. 랜덤 편지로 매칭이 완료된 후, 추가적으로 편지를 주고받게 될 경우 상대방이 보내는 편지들의 배송 현황이 진행 바 형식으로 표시되며, 여러 건의 편지가 오고 있을 경우, 여러 줄로 표시됩니다. 각 진행 바의 완료는 편지의 도착을 의미합니다. 상대방이 편지를 보냈고 나에게 오고 있다는 사실을 유저에게 알림으로써 편지를 기다리는 시간과 설렘을 느끼게 해주고 싶었습니다.

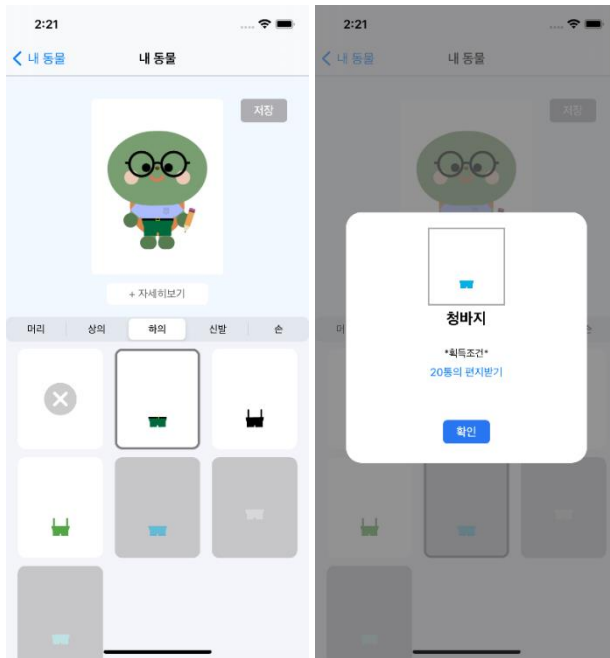
4. 커스텀 화면

4-1. 내 동물 목록



내가 보 하고 있는 모든 동물이 표시됩니다. 동물별로 유저가 꾸며 둔 형태로 표시되며, 동물 별 배송시간 정보가 함께 표시되고, 꾸미고자 하는 동물을 클릭 시 꾸미기 화면으로 이동할 수 있습니다.

4-2. 커스터마이징



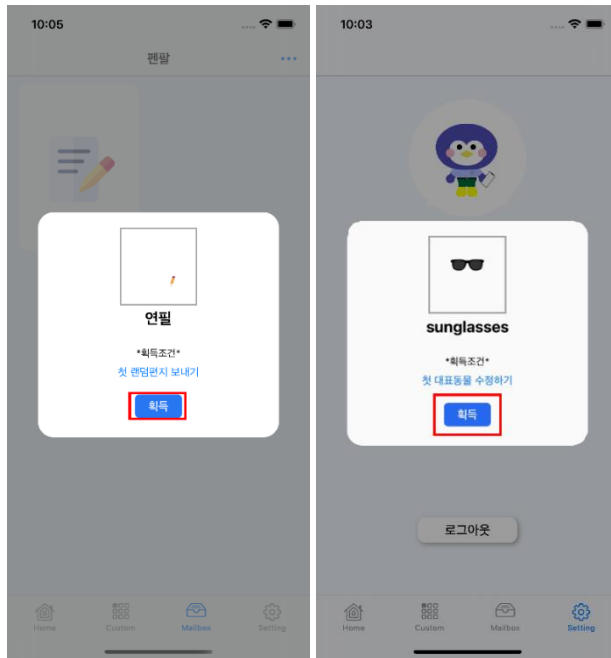
꾸미고자 하는 동물을 클릭 시 커스터마이징 화면으로 이동합니다.

각 동물별로 보유한 액세서리들을 착용할 수 있으며, 첫 번째 x 표시를 클릭 시 해당 부위의 액세서리 착용을 해제합니다.

보유하지 않은 액세서리는 회색으로 표시되며, 클릭 시 해당 액세서리를 획득할 수 있는 획득 조건을 확인할 수 있습니다.

원하는 대로 꾸며준 후 오른쪽 상단의 저장 버튼을 누르면 해당 모습으로 저장됩니다.

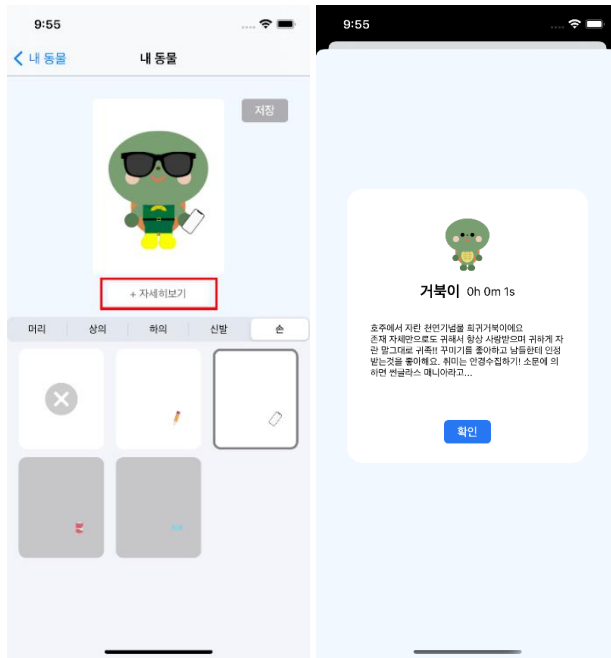
4-3. 액세서리 획득



액세서리 별로 획득 조건을 달성하면 액세서리를 획득했음을 알리는 알림 창과 함께 해당 액세서리 사용이 활성화됩니다.

조건을 달성하는 순간 획득 안내 화면이 등장하고 획득 버튼을 누르면 이후 커스텀 화면으로 이동시 해당 액세서리가 활성화 되어있어 꾸미는데 자유롭게 사용 가능합니다.

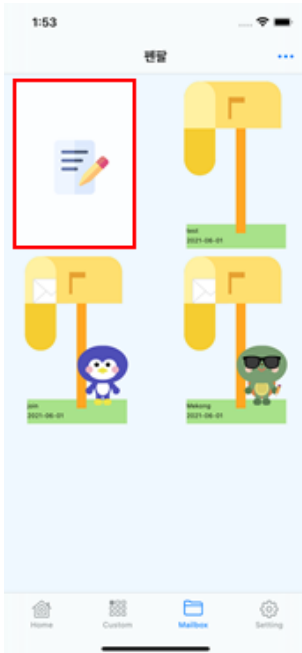
4-4. 동물 스토리 확인



동물별로 자세히 보기 버튼을 클릭하면 해당 동물의 스토리를 확인할 수 있습니다. 각 동물별로 스토리를 설정하여 게임적인 요소를 가미했고, 추후 스토리와 부합하는 액세서리 착용시 특별 효과 적용 등의 추가적인 재미요소를 기획하고 있습니다.

5. 편지함

5-1. 편지 목록 및 랜덤 편지 발송



내가 주고받고 있는 편지함들이 리스트 형식으로 표출됩니다.

편지함은 상대 유저당 1 개씩 생성되며, 상대방의 아이디와 최근 수신 날짜가 표시됩니다. 아직 읽지 않은 편지가 있을 경우 편지를 들고 온 동물의 모습과 함께 편지함에 편지 봉투가 들어있게 됩니다. 편지를 읽게 되면 동물은 배송을 마치고 주인에게 돌아가므로 사라지고, 편지 모양 또한 사라집니다.

동물과 편지 이미지의 유무로 새로 도착한 편지가 있음을 한눈에 확인할 수 있습니다.

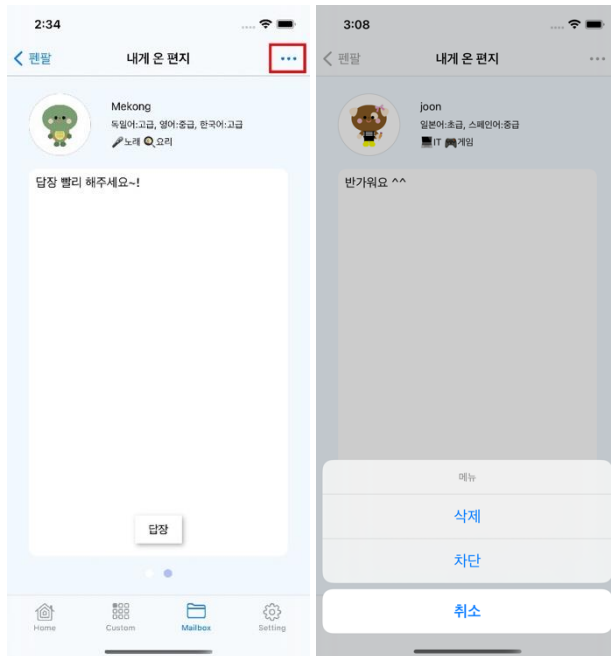
5-2. 랜덤 편지

편지함 리스트의 첫번째 편지 작성 모양 셀을 클릭하여 랜덤 편지를 보낼 수 있습니다.

편지의 작성 방식은 앞서 답장하기와 동일하게 내용 및 배송할 동물을 선택한 후 발송하게 됩니다. 편지의 발송에는 동물들에 적용 되어있는 전송 시간이 소요되며, 해당 시간이 지난 후 매칭된 사람들의 메인 화면에 노출됩니다.

누군가가 해당 편지에 답장을 하게 되면 편지함에 해당 유저의 편지함이 생성되며 편지가 도착하게 됩니다.

5-3. 편지 답장 및 삭제, 차단



편지함을 클릭 시 해당 유저와 주고받은 편지들을 볼 수 있으며, 좌우 스와이프 기능을 통해 편지들을 넘길 수 있습니다.

상대방이 보낸 편지에는 답장 버튼이 생성되어 있어 클릭을 통해 답장하기로 이동할 수 있습니다.

답장의 방식은 앞서 나왔던 편지의 전송 방식과 동일합니다.

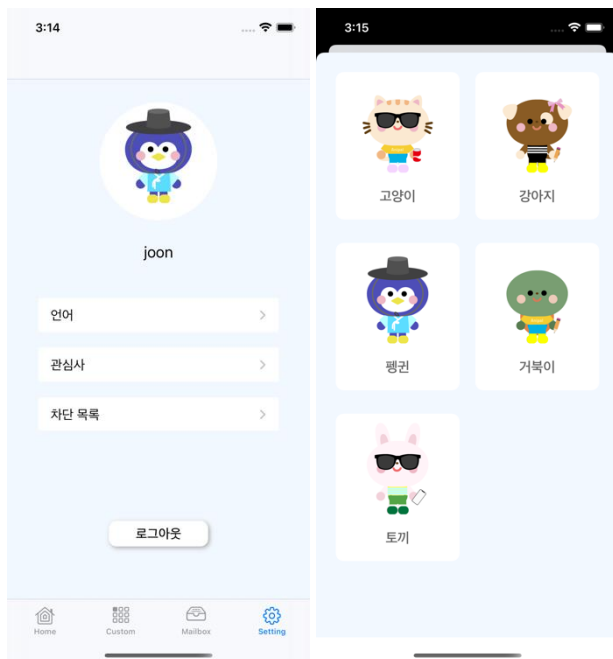
오른쪽 상단의 ... 버튼을 눌러 편지함의 삭제 및 유저 차단을 할 수 있습니다.

삭제를 하게 되면 편지함이 삭제되며 상대방에게는 더이상 답장 버튼이 활성화되지 않아 추가적인 답장을 이어갈 수 없습니다. 이 경우에는 차단한 것이 아니라 단순히 대화방을 나간 것 이기 때문에, 추후에 다시 랜덤 편지를 통해 매칭이 된다면 대화를 이어 갈 수 있습니다.

차단을 하게 되면 해당 유저를 차단목록으로 이동시키고 더이상 해당 유저와는 매칭이 되지도 않고, 답장을 받지 않습니다. 내가 차단을 풀기 전까지 상대와는 절대 이어지지 않습니다.

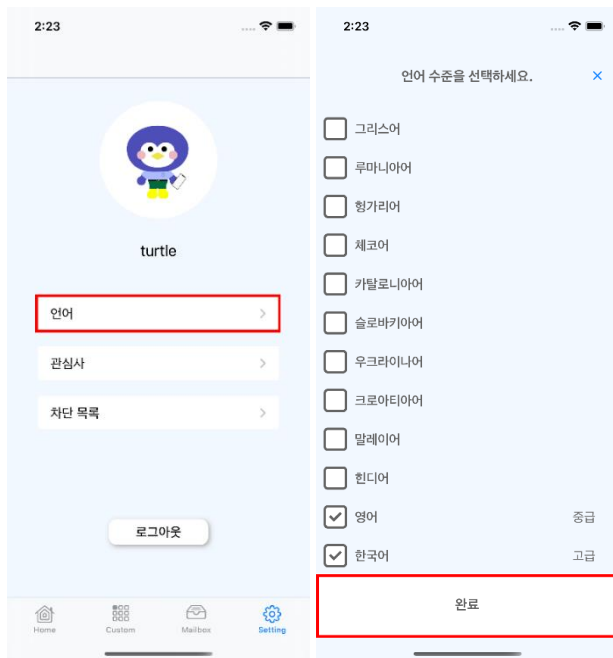
6. 환경 설정

6-1. 대표 동물 수정



환경설정 탭에 떠있는 동그란 동물 모양을 클릭하면 나의 대표 동물을 수정할 수 있습니다. 보유한 동물 중 가장 마음에 드는 동물을 예쁘게 꾸며 대표 동물로 설정해 둘 수 있으며, 환경설정 탭 및 추후 프로필 화면 제작 시 나를 나타낼 동물로 사용될 예정입니다. 언제든지 변경 가능하며, 커스텀 기능을 통해 동물을 새롭게 꾸밀 경우 환경 설정 탭도 그에 맞게 업데이트 됩니다.

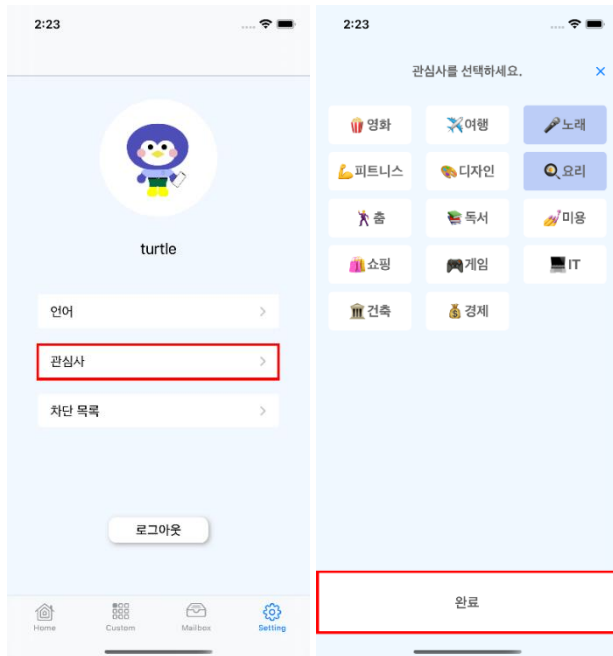
6-2. 언어 수정



회원 가입시 설정한 언어 정보를 언제든지 수정할 수 있습니다.

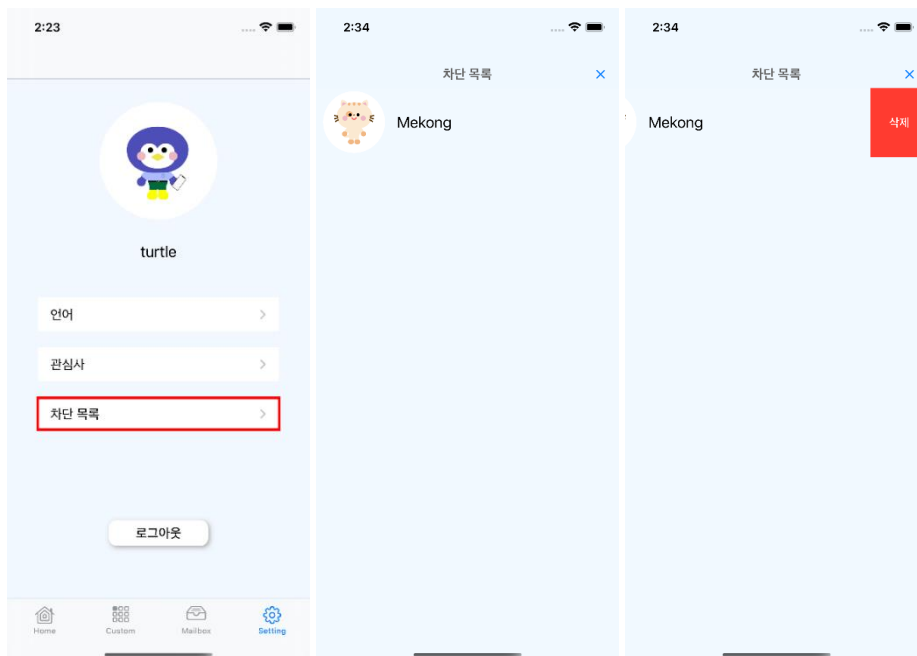
의도치 않게 잘못 설정되었거나, 추가적인 언어 학습을 통해 상태가 변경되었다면 언제든지 수정해서 나의 언어사용 수준을 알릴 수 있습니다.

6-3. 관심사 수정



관심사 역시 회원가입 할 시점과 달라졌다면, 언제든지 환경설정 탭을 통하여 수정 가능합니다.

6-4. 차단 목록



차단 목록을 통해 차단한 유저들을 확인할 수 있으며, 차단 해제를 통해 원하는 유저의 차단을 해제할 수도 있습니다.

차단 해제 시 해당 유저와는 다시 정상적으로 매칭이 이루어집니다.

7. 로컬라이징

세계 시장을 타겟으로 한 펜팔 애플리케이션답게 다중 언어를 지원합니다.

휴대폰 기기의 언어 설정에 따라서 로컬라이즈 한 언어로 표시되며, 현재 한국어와 영어에 대해서 로컬라이징이 완료된 상태입니다.

지원 가능 언어는 추후 중국어, 일본어를 시작으로 더 많은 언어들로 확장해 나갈 계획입니다.

Back-end

개발 환경

- TypeScript
- Node.js
- Nest.js
- MongoDB (mongoose)
- AWS EC2
- AWS S3
- MongoDB Atlas Database
- Docker
- Nginx

구현 사항

API 서버

- Node.js, Nest.js 를 사용하여 웹서버 개발 및 구축
- AWS EC2 배포
- docker-compose 를 활용하여 nginx 배포

```
version: "3.7"
services:
  web:
    image: nginx:stable
    volumes:
      - ./nginx/nginx.conf:/tmp/nginx.conf
      - /var/www/anipal:/var/www/anipal
    environment:
      - NEST_SERVER_ADDR=backend:9091
    command: /bin/bash -c "envsubst < /tmp/nginx.conf > /etc/nginx/conf.d/default.conf && nginx -g 'daemon off;'"
    ports:
      - 80:80
      - 443:443
    depends_on:
      - backend
```

```

backend:
  build: nestjs
  environment:
    - NEST_SERVER_PORT=
    - ANIPAL_GOOGLE_CLIENT_ID=
    - ANIPAL_GOOGLE_SECRET_KEY=
    - ANIPAL_FACEBOOK_APP_KEY=
    - ANIPAL_FACEBOOK_SECRET_KEY=
    - MONGO_DATABASE=
    - MONGO_PASSWORD=
    - MONGO_USER=
    - MONGO_HOST=
  volumes:
    - ./nestjs:/src
    - /src/node_modules
  command:
    - bash
    - -c
    - |
      npm run start:dev

```

- nginx 를 활용하여 https 설정 및 http 리다이렉트

```

server {
    listen 80;
    listen [::]:80;
    return 301 https://anipal.co.kr/$request_uri;
}
server {
    listen 443 ssl http2;
    server_name www.anipal.co.kr;

    ssl_certificate /var/www/anipal/fullchain1.pem;
    ssl_certificate_key /var/www/anipal/privkey1.pem;

    ssl_session_timeout 5m;
    ssl_protocols SSLv2 SSLv3 TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;

    location / {
        proxy_pass http://$NEST_SERVER_ADDR;
    }
}

```


DB 서버

- MongoDB Atlas Database 를 활용하여 클라우드 DB 서버 구축
- Node.js 에서 MongoDB 를 활용하기 위해 mongoose module 사용

이미지 서버

- AWS S3 를 이용해 동물 및 액세서리 사진 저장

API 명세서

Auth

- Refresh

Method: GET

URL: /auth/refresh

Authorizations : None

Request body : None

Response

- Success

HTTP status code : 200 OK

Response Body 는 따로 없습니다. 대신 Cookie 에 Access Token 과 Refresh Token 이 재 발급됩니다.

- Fail

HTTP status code : 401 Unauthorized

```
{
  "status": 401,
  "message": "Unauthorized"
}
```

name	type	description
status	number	HTTP status code(에러 상황에 따라 변할 수 있습니다.)
message	string	에러 메시지(메시지 내용은 에러 상황에 따라 변할 수 있습니다.)

- Login

Method : GET

URL : /auth/:provider

Authorizations : None

URL Parameters

name	type	description	example
provider	string	소셜 로그인 공급 업체	google

Request body : None

Response

name	type	description
_id	string	고유 id 값
name	string	이름
age	number	나이
email	string	이메일
country	string	국적
birthday	string	생년월일
gender	string	성별
favorites	string[]	관심사
banned_users_id	string[]	차단한 유저 id 들
languages	object[]	사용 가능한 언어 객체
languages - name	string	언어 이름
languages - level	string	언어 숙련도
favorite_animal	object	대표 동물 객체
favorite_animal - animal_url	string	대표 동물 이미지 url
favorite_animal - head_url	string	대표 동물이 착용한 모자 이미지 url
favorite_animal - top_url	string	대표 동물이 착용한 상의 이미지 url

favorite_animal - pants_url	string	대표 동물이 착용한 바지 이미지 url
favorite_animal - shoes_url	string	대표 동물이 착용한 신발 이미지 url
favorite_animal - gloves_url	string	대표 동물이 착용한 장갑 이미지 url

- Success

HTTP status code : 200 OK

```
{
  "_id": "60681268ab645e32c4191cad",
  "name": "캡스톤",
  "age": 20,
  "email": "capstone@gamil.com",
  "country": "Korean",
  "birthday": "2020-01-10",
  "gender": "female",
  "favorites": [
    "Movie",
    "Dance"
  ],
  "banned_users_id": [
    "60922496b58ba001327134d6"
  ],
  "languages": [
    {
      "name": "ja",
      "level": 0
    }
  ],
  "favorite_animal": {
    "animal_url": "https://s3.domain/filename.png",
    "head_url": "https://s3.domain/filename.png",
    "top_url": "https://s3.domain/filename.png",
    "pants_url": "https://s3.domain/filename.png",
    "shoes_url": "https://s3.domain/filename.png",
    "gloves_url": "https://s3.domain/filename.png"
  }
}
```

- Fail

HTTP status code : 400 Bad Request

```
{
  "status": 400,
  "message": "email 과 같은 이메일을 가진 다른 소셜 계정이 존재합니
```

```
다."
}
```

HTTP status code : 404 Not found

```
{
  "status": 404,
  "message": "email 에 해당하는 유저가 존재하지 않습니다."
}
```

name	type	description
status	number	HTTP status code(에러 상황에 따라 변할 수 있습니다.)
message	string	에러 메시지(메시지 내용은 에러 상황에 따라 변할 수 있습니다.)

- Register

Method : POST

URL : /auth/register

Authorizations : None

Request body

name	type	require	description
name	string	true	이름
gender	string	true	성별
age	number	true	나이
birthday	string	true	생년월일
email	string	false	이메일
provider	string	true	소셜 로그인 업체
favorites	string[]	true	관심사
languages	object[]	true	사용 가능한 언어

languages - name	string	true	언어 이름
languages - level	number	true	언어 숙련도
favorite_animal	string	true	대표 동물 id

Request Body Example

```
{
  "name": "캡스톤",
  "gender": "male",
  "age": 30,
  "birthday": "2020-01-10",
  "country": "Korea",
  "email": "capstone@gmail.com",
  "provider": "google",
  "favorites": [
    "Movie",
    "Dance"
  ],
  "languages": [
    {
      "name": "ja",
      "level": 0
    }
  ],
  "favorite_animal": "60681268ab645e32c4191cad"
}
```

Response

name	type	description
_id	string	고유 id 값
name	string	이름
age	number	나이
email	string	이메일
country	string	국적
birthday	string	생년월일
gender	string	성별
favorites	string[]	관심사

banned_users_id	string[]	차단한 유저 id 들
languages	object[]	사용 가능한 언어 객체
languages - name	string	언어 이름
languages - level	string	언어 숙련도
favorite_animal	object	대표 동물 객체
favorite_animal - animal_url	string	대표 동물 이미지 url
favorite_animal - head_url	string	대표 동물이 착용한 모자 이미지 url
favorite_animal - top_url	string	대표 동물이 착용한 상의 이미지 url
favorite_animal - pants_url	string	대표 동물이 착용한 바지 이미지 url
favorite_animal - shoes_url	string	대표 동물이 착용한 신발 이미지 url
favorite_animal - gloves_url	string	대표 동물이 착용한 장갑 이미지 url

- Success

HTTP status code : 201 Created

```
{
  "_id": "60681268ab645e32c4191cad",
  "name": "캡스톤",
  "age": 20,
  "email": "capstone@gamil.com",
  "country": "Korean",
  "birthday": "2020-01-10",
  "gender": "string",
  "favorites": [
    "Movie",
    "Dance"
  ],
  "banned_users_id": [
    "60922496b58ba001327134d6"
  ],
  "languages": [
    {
      "name": "ja",
      "level": 0
    }
  ],
  "favorite_animal": {
    "animal_url": "https://s3.domain/filename.png",
```

```

        "head_url": "https://s3.domain/filename.png",
        "top_url": "https://s3.domain/filename.png",
        "pants_url": "https://s3.domain/filename.png",
        "shoes_url": "https://s3.domain/filename.png",
        "gloves_url": "https://s3.domain/filename.png"
    }
}

```

– Fail

HTTP status code : 400 Bad Request

```

{
  "status": 400,
  "message": "email 과 같은 이메일을 가진 계정이 이미 존재합니다."
}

```

name	type	description
status	number	HTTP status code(에러 상황에 따라 변할 수 있습니다.)
message	string	에러 메시지(메시지 내용은 에러 상황에 따라 변할 수 있습니다.)

Users

- Get all users

Method : GET

URL : /users

Authorizations : JWT

Request body : None

Response

name	type	description
_id	string	고유 id 값
name	string	이름
age	number	나이

email	string	이메일
country	string	국적
birthday	string	생년월일
gender	string	성별
favorites	string[]	관심사
banned_users_id	string[]	차단한 유저 id 들
languages	object[]	사용 가능한 언어 객체
languages - name	string	언어 이름
languages - level	string	언어 숙련도
favorite_animal	object	대표 동물 객체
favorite_animal - animal_url	string	대표 동물 이미지 url
favorite_animal - head_url	string	대표 동물이 착용한 모자 이미지 url
favorite_animal - top_url	string	대표 동물이 착용한 상의 이미지 url
favorite_animal - pants_url	string	대표 동물이 착용한 바지 이미지 url
favorite_animal - shoes_url	string	대표 동물이 착용한 신발 이미지 url
favorite_animal - gloves_url	string	대표 동물이 착용한 장갑 이미지 url

- Success

HTTP status code : 200 OK

```
[
  {
    "_id": "60681268ab645e32c4191cad",
    "name": "캡스톤",
    "age": 20,
    "email": "capstone@gamil.com",
    "country": "Korean",
    "birthday": "2020-01-10",
    "gender": "female",
    "favorites": [
      "Movie",
      "Dance"
    ],
  },
]
```

```

    "banned_users_id": [
      "60922496b58ba001327134d6"
    ],
    "languages": [
      {
        "name": "ja",
        "level": 0
      }
    ],
    "favorite_animal": {
      "animal_url": "https://s3.domain/filename.png",
      "head_url": "https://s3.domain/filename.png",
      "top_url": "https://s3.domain/filename.png",
      "pants_url": "https://s3.domain/filename.png",
      "shoes_url": "https://s3.domain/filename.png",
      "gloves_url": "https://s3.domain/filename.png"
    }
  }
]

```

- Get one user

Method : GET

URL : /users/:userId

Authorizations : JWT

URL Parameters

name	type	description	example
userId	string	유저의 id	60681268ab645e32c4191cad

Request body : None

Response

name	type	description
_id	string	고유 id 값
name	string	이름
age	number	나이
email	string	이메일
country	string	국적

birthday	string	생년월일
gender	string	성별
favorites	string[]	관심사
banned_users_id	string[]	차단한 유저 id 들
languages	object[]	사용 가능한 언어 객체
languages - name	string	언어 이름
languages - level	string	언어 숙련도
favorite_animal	object	대표 동물 객체
favorite_animal - animal_url	string	대표 동물 이미지 url
favorite_animal - head_url	string	대표 동물이 착용한 모자 이미지 url
favorite_animal - top_url	string	대표 동물이 착용한 상의 이미지 url
favorite_animal - pants_url	string	대표 동물이 착용한 바지 이미지 url
favorite_animal - shoes_url	string	대표 동물이 착용한 신발 이미지 url
favorite_animal - gloves_url	string	대표 동물이 착용한 장갑 이미지 url

- Success

HTTP status code : 200 OK

```
{
  "_id": "60681268ab645e32c4191cad",
  "name": "캡스톤",
  "age": 20,
  "email": "capstone@gamil.com",
  "country": "Korean",
  "birthday": "2020-01-10",
  "gender": "female",
  "favorites": [
    "Movie",
    "Dance"
  ],
  "banned_users_id": [
    "60922496b58ba001327134d6"
  ],
  "languages": [
    {
```

```

        "name": "ja",
        "level": 0
    }
],
"favorite_animal": {
    "animal_url": "https://s3.domain/filename.png",
    "head_url": "https://s3.domain/filename.png",
    "top_url": "https://s3.domain/filename.png",
    "pants_url": "https://s3.domain/filename.png",
    "shoes_url": "https://s3.domain/filename.png",
    "gloves_url": "https://s3.domain/filename.png"
}
}

```

- Fail

HTTP status code : 400 Bad Request

```

{ "status": 400,    "message": "id 에 해당하는 유저가 존재하지 않습니다."}

```

name	type	description
status	number	HTTP status code(에러 상황에 따라 변할 수 있습니다.)
message	string	에러 메시지(메시지 내용은 에러 상황에 따라 변할 수 있습니다.)

- Update user profile

Method : POST

URL : /users/:userId

URL Parameters

name	type	description	example
userId	string	유저의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body

name	type	require	description
------	------	---------	-------------

name	string	false	이름
favorites	string[]	false	관심사
languages	object[]	false	사용 가능한 언어
languages - name	string	false	언어 이름
languages - level	number	false	언어 숙련도
favorite_animal	object	false	대표 동물 객체
favorite_animal - animal_url	string	false	대표 동물의 이미지 url
favorite_animal - head_url	string	false	대표 동물이 착용한 모자 이미지 url
favorite_animal - top_url	string	false	대표 동물이 착용한 상의 이미지 url
favorite_animal - pants_url	string	false	대표 동물이 착용한 바지 이미지 url
favorite_animal - shoes_url	string	false	대표 동물이 착용한 신발 이미지 url
favorite_animal - gloves_url	string	false	대표 동물이 착용한 장갑 이미지 url

Request Body Example

```
{
  "name": "캡스톤",
  "favorites": [
    "Movie",
    "Dance"
  ],
  "languages": [
    {
      "name": "ja",
      "level": 0
    }
  ],
  "favorite_animal": {
    "animal_url": "https://s3.domain/filename.png",
```

```

    "head_url": "https://s3.domain/filename.png",
    "top_url": "https://s3.domain/filename.png",
    "pants_url": "https://s3.domain/filename.png",
    "shoes_url": "https://s3.domain/filename.png",
    "gloves_url": "https://s3.domain/filename.png"
  }
}

```

Response

name	type	description
_id	string	고유 id 값
name	string	이름
age	number	나이
email	string	이메일
country	string	국적
birthday	string	생년월일
gender	string	성별
favorites	string[]	관심사
banned_users_id	string[]	차단한 유저 id 들
languages	object[]	사용 가능한 언어 객체
languages - name	string	언어 이름
languages - level	string	언어 숙련도
favorite_animal	object	대표 동물 객체
favorite_animal - animal_url	string	대표 동물 이미지 url
favorite_animal - head_url	string	대표 동물이 착용한 모자 이미지 url
favorite_animal - top_url	string	대표 동물이 착용한 상의 이미지 url
favorite_animal - pants_url	string	대표 동물이 착용한 바지 이미지 url
favorite_animal - shoes_url	string	대표 동물이 착용한 신발 이미지 url
favorite_animal - gloves_url	string	대표 동물이 착용한 장갑 이미지 url

- Success

HTTP status code : 201 Created

```
{
  "_id": "60681268ab645e32c4191cad",
  "name": "캡스톤",
  "age": 20,
  "email": "capstone@gamil.com",
  "country": "Korean",
  "birthday": "2020-01-10",
  "gender": "string",
  "favorites": [
    "Movie",
    "Dance"
  ],
  "banned_users_id": [
    "60922496b58ba001327134d6"
  ],
  "languages": [
    {
      "name": "ja",
      "level": 0
    }
  ],
  "favorite_animal": {
    "animal_url": "https://s3.domain/filename.png",
    "head_url": "https://s3.domain/filename.png",
    "top_url": "https://s3.domain/filename.png",
    "pants_url": "https://s3.domain/filename.png",
    "shoes_url": "https://s3.domain/filename.png",
    "gloves_url": "https://s3.domain/filename.png"
  }
}
```

- Fail

HTTP status code : 400 Bad Request

```
{
  "status": 400,
  "message": "id 에 해당하는 유저가 존재하지 않습니다."
}
```

name	type	description
status	number	HTTP status code(에러 상황에 따라 변할 수 있습니다.)

message

string

에러 메시지(메시지 내용은
에러 상황에 따라 변할 수
있습니다.)

- Block other user

Method: PUT

URL: /users/ban/:bannedUserId

URL Parameters

name	type	description	example
bannedUserId	string	차단할 유저의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body : None

Response

- Success

HTTP status code : 200 OK

Response Body 는 따로 없습니다.

- Unblock other user

Method: DELETE

URL: /users/ban/:bannedUserId

URL Parameters

name	type	description	example
bannedUserId	string	차단할 유저의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body : None

Response

- Success

HTTP status code : 200 OK

Response Body 는 따로 없습니다.

Owns

- Get all my animals

Method: GET

URL: /own/animals

Authorizations : JWT

Request body : None

Response

name	type	description
_id	string	고유 id 값
animal	object	동물 정보 객체
animal - localized	string	앱에서 localizing 에 필요한 key value
animal - delay_time	string	동물의 편지 전송 시간
animal_url	string	동물 이미지 url
head_url	string	동물이 착용한 모자 이미지 url
top_url	string	동물이 착용한 상의 이미지 url
pants_url	string	동물이 착용한 바지 이미지 url
shoes_url	string	동물이 착용한 신발 이미지 url
gloves_url	string	동물이 착용한 장갑 이미지 url
is_used	boolean	동물이 현재 편지 전송 중인지 여부
delay_time	boolean	동물의 편지 전송 시간
coming_animal	object	동물이 전송 중일 때 표시되는 이미지 객체 정보
coming_animal - animal_url	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 표시되는 이미지 url

coming_animal - bar	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 표시되는 bar 의 색깔 값
coming_animal - background	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 background 의 색깔 값
is_favorite	boolean	대표 동물인지 여부

- Success

HTTP status code : 200 OK

```
[
  {
    "_id": "60680be281b87e19745e7f0a",
    "animal": {
      "localized": "cat",
      "delay_time": "1h 0m 0s"
    },
    "animal_url": "https://s3.domain/filename.png",
    "head_url": "https://s3.domain/filename.png",
    "top_url": "https://s3.domain/filename.png",
    "pants_url": "https://s3.domain/filename.png",
    "shoes_url": "https://s3.domain/filename.png",
    "gloves_url": "https://s3.domain/filename.png",
    "is_used": true,
    "delay_time": "1h 0m 0s",
    "coming_animal": {
      "animal_url": "https://aws.s3.domain.com/filename.png",
      "bar": "090909",
      "background": "090909"
    },
    "is_favorite": false
  }
]
```

- Customizing my animal

Method: PUT

URL: /own/animals/:ownAnimalId

URL Parameters

name	type	description	example
ownAnimalId	string	커스터 마이징 할 내 동물의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body

name	type	require	description
head_url	string	false	동물이 착용할 모자 이미지 url
top_url	string	false	동물이 착용할 상의 이미지 url
pants_url	string	false	동물이 착용할 바지 이미지 url
shoes_url	string	false	동물이 착용할 신발 이미지 url
gloves_url	string	false	동물이 착용할 장갑 이미지 url

Response

name	type	description
_id	string	고유 id 값
animal	object	동물 정보 객체
animal - localized	string	앱에서 localizing 에 필요한 key value
animal - delay_time	string	동물의 편지 전송 시간
animal_url	string	동물 이미지 url
head_url	string	동물이 착용한 모자 이미지 url
top_url	string	동물이 착용한 상의 이미지 url
pants_url	string	동물이 착용한 바지 이미지 url
shoes_url	string	동물이 착용한 신발 이미지 url
gloves_url	string	동물이 착용한 장갑 이미지 url
is_used	boolean	동물이 현재 편지 전송 중인지 여부
delay_time	boolean	동물의 편지 전송 시간
coming_animal	object	동물이 전송 중일 때 표시되는 이미지 객체 정보
coming_animal - animal_url	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 표시되는 이미지 url

coming_animal - bar	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 표시되는 bar 의 색깔 값
coming_animal - background	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 background 의 색깔 값
is_favorite	boolean	대표 동물인지 여부

- Success

HTTP status code : 200 OK

```
{
  "_id": "60680be281b87e19745e7f0a",
  "animal": {
    "localized": "cat",
    "delay_time": "1h 0m 0s"
  },
  "animal_url": "https://s3.domain/filename.png",
  "head_url": "https://s3.domain/filename.png",
  "top_url": "https://s3.domain/filename.png",
  "pants_url": "https://s3.domain/filename.png",
  "shoes_url": "https://s3.domain/filename.png",
  "gloves_url": "https://s3.domain/filename.png",
  "is_used": true,
  "delay_time": "1h 0m 0s",
  "coming_animal": {
    "animal_url": "https://aws.s3.domain.com/filename.png",

    "bar": "090909",
    "background": "090909"
  },
  "is_favorite": false
}
```

- Fail

HTTP status code : 400 Bad Request

```
{ "status": 400,    "message": "id 에 해당하는 유저가 존재하지 않습니다." }
```

name	type	description
status	number	HTTP status code(에러 상황에 따라 변할 수 있습니다.)

message

string

에러 메시지(메시지 내용은
에러 상황에 따라 변할 수
있습니다.)

Letters

- Get all letters coming me

Method: GET

URL: /letters/coming

Authorizations : JWT

Request body : None

Response

name	type	description
arrive_time	Date	도착 시간
send_time	Date	전송 시간
sender_name	string	송신자 이름
coming_animal	object	동물이 전송 중일 때 표시되는 이미지 객체 정보
coming_animal - animal_url	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 표시되는 이미지 url
coming_animal - bar	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 표시되는 bar 의 색깔 값
coming_animal - background	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 background 의 색깔 값

- Success

HTTP status code : 200 OK

```
[  
  {  
    "arrive_time": "2021-04-04T16:25:57.075Z",  
    "send_time": "2021-04-04T16:25:57.075Z",
```

```

        "sender_name": "isabella",
        "coming_animal": {
            "animal_url": "https://aws.s3.domain.com/filename.png",
            "bar": "090909",
            "background": "090909"
        }
    }
]

```

- Get all random letters I can see

Method: GET

URL: /letters/random

Authorizations : JWT

Request body : None

Response

name	type	description
_id	string	고유 id 값
post_animal	object	편지를 전송 중인 동물의 객체 정보
post_animal- animal_url	string	동물의 이미지 url
post_animal- head_url	string	동물이 착용한 모자 이미지 url
post_animal- top_url	string	동물이 착용한 상의 이미지 url
post_animal- pants_url	string	동물이 착용한 바지 이미지 url
post_animal- shoes_url	string	동물이 착용한 신발 이미지 url
post_animal - gloves_url	string	동물이 착용한 장갑 이미지 url

- Success

HTTP status code : 200 OK

```

[
  {
    "_id": "60680be281b87e19745e7f0a",
    "post_animal": {
      "animal_url": "https://s3.domain/filename.png",
      "head_url": "https://s3.domain/filename.png",
      "top_url": "https://s3.domain/filename.png",
      "pants_url": "https://s3.domain/filename.png",

```

```

        "shoes_url": "https://s3.domain/filename.png",
        "gloves_url": "https://s3.domain/filename.png"
    }
}
]

```

- Get one random letter selected

Method: GET

URL: /letters/random/:randomLetterId

URL Parameters

name	type	description	example
randomLetterId	string	랜덤 편지의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body : None

Response

name	type	description
_id	string	고유 id 값
content	string	내용
send_time	Date	전송된 시간
arrive_time	Date	도착 시간
sender	object	편지를 보낸 유저의 객체 정보
sender- user_id	string	유저의 고유 id
sender- name	string	유저의 이름
sender- country	string	유저의 국적
sender- favorites	string[]	유저의 관심사들
sender- languages	object[]	유저가 사용 가능한 언어 객체 정보
sender- languages - name	string	언어 이름
sender - languages - level	number	언어 숙련도

post_animal	object	편지를 전송 중인 동물의 객체 정보
post_animal- animal_url	string	동물의 이미지 url
post_animal- head_url	string	동물이 착용한 모자 이미지 url
post_animal- top_url	string	동물이 착용한 상의 이미지 url
post_animal- pants_url	string	동물이 착용한 바지 이미지 url
post_animal- shoes_url	string	동물이 착용한 신발 이미지 url
post_animal - gloves_url	string	동물이 착용한 장갑 이미지 url

- Success

HTTP status code : 200 OK

```
{
  "_id": "6071ad8d4f29c9d6f5393309",
  "content": "Help me, please~",
  "send_time": "2021-04-04T14:25:57.075Z",
  "arrive_time": "2021-04-04T16:25:57.075Z",
  "sender": {
    "user_id": "6071ad8d4f29c9d6f5393309",
    "name": "isabella",
    "country": "Korea",
    "favorites": [
      "Movie"
    ],
    "languages": [
      {
        "name": "ja",
        "level": 0
      }
    ]
  },
  "post_animal": {
    "animal_url": "https://s3.domain/filename.png",
    "head_url": "https://s3.domain/filename.png",
    "top_url": "https://s3.domain/filename.png",
    "pants_url": "https://s3.domain/filename.png",
    "shoes_url": "https://s3.domain/filename.png",
    "gloves_url": "https://s3.domain/filename.png"
  }
}
```


- Get one letter selected

Method: GET

URL: /letters/:letterId

URL Parameters

name	type	description	example
letterId	string	편지의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body : None

Response

name	type	description
_id	string	고유 id 값
content	string	내용
send_time	Date	전송된 시간
arrive_time	Date	도착 시간
sender	object	편지를 보낸 유저의 객체 정보
sender- user_id	string	유저의 고유 id
sender- name	string	유저의 이름
sender- country	string	유저의 국적
sender- favorites	string[]	유저의 관심사들
sender- languages	object[]	유저가 사용 가능한 언어 객체 정보
sender- languages - name	string	언어 이름
sender - languages - level	number	언어 숙련도
post_animal	object	편지를 전송 중인 동물의 객체 정보
post_animal- animal_url	string	동물의 이미지 url
post_animal- head_url	string	동물이 착용한 모자 이미지 url
post_animal- top_url	string	동물이 착용한 상의 이미지 url

post_animal- pants_url	string	동물이 착용한 바지 이미지 url
post_animal- shoes_url	string	동물이 착용한 신발 이미지 url
post_animal - gloves_url	string	동물이 착용한 장갑 이미지 url

- Success

HTTP status code : 200 OK

```
{
  "_id": "6071ad8d4f29c9d6f5393309",
  "content": "Help me, please~",
  "send_time": "2021-04-04T14:25:57.075Z",
  "arrive_time": "2021-04-04T16:25:57.075Z",
  "sender": {
    "user_id": "6071ad8d4f29c9d6f5393309",
    "name": "isabella",
    "country": "Korea",
    "favorites": [
      "Movie"
    ],
    "languages": [
      {
        "name": "ja",
        "level": 0
      }
    ]
  },
  "post_animal": {
    "animal_url": "https://s3.domain/filename.png",
    "head_url": "https://s3.domain/filename.png",
    "top_url": "https://s3.domain/filename.png",
    "pants_url": "https://s3.domain/filename.png",
    "shoes_url": "https://s3.domain/filename.png",
    "gloves_url": "https://s3.domain/filename.png"
  }
}
```

- Fail

HTTP status code : 400 Bad Request

```
{
  "status": 400,
  "message": "해당 편지의 수신자가 아닙니다."
}
```

name	type	description
status	number	HTTP status code(에러 상황에 따라 변할 수 있습니다.)
message	string	에러 메시지(메시지 내용은 에러 상황에 따라 변할 수 있습니다.)

- **Post random letter**

Method: POST

URL: /letters/random

Authorizations : JWT

Request body

name	type	require	description
content	string	true	편지 내용
own_animal_id	string	true	편지를 전송할 유저의 동물 id

Response

name	type	description
_id	string	고유 id 값
post_animal	object	편지를 전송 중인 동물의 객체 정보
post_animal- animal_url	string	동물의 이미지 url
post_animal- head_url	string	동물이 착용한 모자 이미지 url
post_animal- top_url	string	동물이 착용한 상의 이미지 url
post_animal- pants_url	string	동물이 착용한 바지 이미지 url
post_animal- shoes_url	string	동물이 착용한 신발 이미지 url
post_animal - gloves_url	string	동물이 착용한 장갑 이미지 url

- Success

HTTP status code : 201 Created

```

{
  "_id": "60680be281b87e19745e7f0a",
  "post_animal": {
    "animal_url": "https://s3.domain/filename.png",
    "head_url": "https://s3.domain/filename.png",
    "top_url": "https://s3.domain/filename.png",
    "pants_url": "https://s3.domain/filename.png",
    "shoes_url": "https://s3.domain/filename.png",
    "gloves_url": "https://s3.domain/filename.png"
  }
}

```

- Post letter

Method: POST

URL: /letters

Authorizations : JWT

Request body

name	type	require	description
content	string	true	편지 내용
receiver	string	true	수신 유저의 id
own_animal_id	string	true	편지를 전송할 유저의 동물 id

Response

- Success

HTTP status code : 201 Created

- Reply the random letter

Method: POST

URL: /letters/random/:randomLetterId

URL Parameters

name	type	description	example
randomLetterId	string	랜덤 편지의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body

name	type	require	description
content	string	true	편지 내용
own_animal_id	string	true	편지를 전송할 유저의 동물 id

Response

- Success

HTTP status code : 201 Created

- Fail

HTTP status code : 400 Bad Request

```
{
  "status": 400,
  "message": "존재하지 않는 편지입니다."
}
```

name	type	description
status	number	HTTP status code(에러 상황에 따라 변할 수 있습니다.)
message	string	에러 메시지(메시지 내용은 에러 상황에 따라 변할 수 있습니다.)

- Don't show anymore specific random letter

Method: PUT

URL: /letters/random/:randomLetterId

URL Parameters

name	type	description	example
randomLetterId	string	랜덤 편지의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body : None

Response

- Success

HTTP status code : 200 OK

Request Body 가 없습니다. 해당 작업이 완료되면 지정한 랜덤 편지는
유저에게 더 이상 보이지 않습니다.

Languages

- Get all languages

Method: GET

URL: /languages

Authorizations : None

Request body : None

Response

name	type	description
name	string	언어 이름

- Success

HTTP status code : 200 OK

```
[  
  {  
    "name": "ja"  
  }  
]
```

Favorites

- Get all favorites

Method: GET

URL: /favorites

Authorizations : None

Request body : None

Response

name	type	description
name	string	관심사 이름
- Success		

HTTP status code : 200 OK

```
[
  {
    "name": "Movie"
  }
]
```

Animals

- Get all animals

Method: GET

URL: /animals

Authorizations : JWT

Request body : None

Response

name	type	description
head_url	string	동물이 착용한 모자 이미지 url
top_url	string	동물이 착용한 상의 이미지 url
pants_url	string	동물이 착용한 바지 이미지 url
shoes_url	string	동물이 착용한 신발 이미지 url

gloves_url	string	동물이 착용한 장갑 이미지 url
is_own	boolean	해당 동물을 유저가 소유 중인지 여부

- Success

HTTP status code : 200 OK

```
[
  {
    "animal_url": "https://s3.domain/filename.png",
    "head_url": "https://s3.domain/filename.png",
    "top_url": "https://s3.domain/filename.png",
    "pants_url": "https://s3.domain/filename.png",
    "shoes_url": "https://s3.domain/filename.png",
    "gloves_url": "https://s3.domain/filename.png",
    "is_own": true
  }
]
```

- Get all basic animals

Method: GET

URL: /animals/basic

Authorizations : JWT

Request body : None

Response

name	type	description
_id	string	고유 id 값
price	number	가격
img_url	string	이미지 url
delay_time	string	편지 배송 시간
coming_animal	object	동물이 전송 중일 때 표시되는 이미지 객체 정보
coming_animal - animal_url	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 표시되는 이미지 url
coming_animal - bar	string	해당 동물이 편지를 전송 중일 때 받는

coming_animal - background	string	사람의 화면에 표시되는 bar 의 색깔 값 해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 background 의 색깔 값
is_basic	boolean	기본 동물인지 여부
localized	string	localizing 을 위한 key value

- Success

HTTP status code : 200 OK

```
[
  {
    "_id": "60680be281b87e19745e7f0a",
    "price": 0,
    "img_url": "https://s3.domain/filename.png",
    "delay_time": "1h 0m 0s",
    "coming_animal": {
      "animal_url": "https://aws.s3.domain.com/filename.png",
      "bar": "090909",
      "background": "090909"
    },
    "is_basic": true,
    "localized": "cat"
  }
]
```

- Get one animal

Method: GET

URL: /animals/:animalId

URL Parameters

name	type	description	example
animalId	string	동물의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body : None

Response

name	type	description
_id	string	고유 id 값

price	number	가격
img_url	string	이미지 url
delay_time	string	편지 배송 시간
coming_animal	object	동물이 전송 중일 때 표시되는 이미지 객체 정보
coming_animal - animal_url	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 표시되는 이미지 url
coming_animal - bar	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 표시되는 bar 의 색깔 값
coming_animal - background	string	해당 동물이 편지를 전송 중일 때 받는 사람의 화면에 background 의 색깔 값
is_basic	boolean	기본 동물인지 여부
localized	string	localizing 을 위한 key value

- Success

HTTP status code : 200 OK

```
{
  "_id": "60680be281b87e19745e7f0a",
  "price": 0,
  "img_url": "https://s3.domain/filename.png",
  "delay_time": "1h 0m 0s",
  "mission": "changeFavoriteAnimal",
  "coming_animal": {
    "animal_url": "https://aws.s3.domain.com/filename.png",
    "bar": "090909",
    "background": "090909"
  },
  "is_basic": true,
  "localized": "cat"
}
```

Accessories

- Get all accessories in a specific category

Method: GET

URL: /accessories/all/:category

URL Parameters

name	type	description	example
category	string	액세서리의 카테고리	top

Authorizations : JWT

Request body : None

Response

name	type	description
accessory_id	string	고유 id 값
img_url	string	이미지 url
is_own	boolean	유저가 소유 중인지 여부

- Success

HTTP status code : 200 OK

```
[
  {
    "accessory_id": "6071ad8d4f29c9d6f5393309",
    "img_url": "https://s3.domain/filename.png",
    "is_own": true
  }
]
```

- Get one accessory

Method: GET

URL: /accessories/:accessoryId

URL Parameters

name	type	description	example
------	------	-------------	---------

accessoryId string 액세서리의 id 60681268ab645e32c4191cad

Authorizations : JWT

Request body : None

Response

name	type	description
_id	string	고유 id 값
name	string	이름
price	number	가격
img_url	string	이미지 url
mission	string	액세서리 획득 미션
category	string	카테고리

- Success

HTTP status code : 200 OK

```
{
  "_id": "60680be281b87e19745e7f0a",
  "name": "허름한 상의",
  "price": 0,
  "img_url": "https://s3.domain/filename.png",
  "mission": "postFirstRandomLetter",
  "category": "string"
}
```

Mailboxes

- Get my mailboxes : GET /mailboxes/my

Method: GET

URL: /mailboxes/my

Authorizations : JWT

Request body : None

Response

name	type	description
------	------	-------------

_id	string	고유 id 값
letters_count	number	들어있는 편지 개수
partner	object	mailbox 를 공유하는 유저 객체 정보
partner - user_id	string	상대 유저의 고유 id
partner - name	string	상대 유저의 이름
partner - country	string	상대 유저의 국적
partner - favorites	string[]	상대 유저의 관심사들
partner - languages	object[]	상대 유저가 사용 가능한 언어 객체 정보
partner - languages - name	string	언어 이름
partner - languages - level	number	언어 숙련도
arrive_date	Date	편지함에 편지가 도착할 시간
is_opened	boolean	편지가 새로 도착했을 때 편지함을 열었는지 여부
is_connected	boolean	메일함이 서로 연결되어 있는지 여부
thumbnail_animal	object	편지함에 새로 편지가 도착했을 때 표시될 썸네일 동물 객체 정보
thumbnail_animal - animal_url	string	동물의 이미지 url
thumbnail_animal - head_url	string	동물이 착용한 모자 이미지 url
thumbnail_animal - top_url	string	동물이 착용한 상의 이미지 url
thumbnail_animal - pants_url	string	동물이 착용한 바지 이미지 url
thumbnail_animal - shoes_url	string	동물이 착용한 신발 이미지 url
thumbnail_animal - gloves_url	string	동물이 착용한 장갑 이미지 url
- Success		

HTTP status code : 200 OK

```
[
  {
    "_id": "6071ad8d4f29c9d6f5393309",
    "letters_count": 1,
    "partner": {
      "user_id": "6071ad8d4f29c9d6f5393309",
      "name": "isabella",
      "country": "Korea",
      "favorites": [
        "Movie"
      ],
      "languages": [
        {
          "name": "ja",
          "level": 0
        }
      ]
    },
    "arrive_date": "2021-04-04T14:25:57.075Z",
    "is_opened": true,
    "is_connected": true,
    "thumbnail_animal": {
      "animal_url": "https://s3.domain/filename.png",
      "head_url": "https://s3.domain/filename.png",
      "top_url": "https://s3.domain/filename.png",
      "pants_url": "https://s3.domain/filename.png",
      "shoes_url": "https://s3.domain/filename.png",
      "gloves_url": "https://s3.domain/filename.png"
    }
  }
]
```

- Get all letters in specific mailbox

Method: GET

URL: /mailboxes/show/:mailboxId

URL Parameters

name	type	description	example
mailboxId	string	메일함의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body : None

Response

name	type	description
_id	string	고유 id 값
content	string	내용
send_time	Date	전송된 시간
arrive_time	Date	도착 시간
sender	object	편지를 보낸 유저의 객체 정보
sender- user_id	string	유저의 고유 id
sender- name	string	유저의 이름
sender- country	string	유저의 국적
sender- favorites	string[]	유저의 관심사들
sender- languages	object[]	유저가 사용 가능한 언어 객체 정보
sender- languages - name	string	언어 이름
sender - languages - level	number	언어 숙련도
post_animal	object	편지를 전송 중인 동물의 객체 정보
post_animal- animal_url	string	동물의 이미지 url
post_animal- head_url	string	동물이 착용한 모자 이미지 url
post_animal- top_url	string	동물이 착용한 상의 이미지 url
post_animal- pants_url	string	동물이 착용한 바지 이미지 url
post_animal- shoes_url	string	동물이 착용한 신발 이미지 url
post_animal - gloves_url	string	동물이 착용한 장갑 이미지 url

- Success

HTTP status code : 200 OK

```
[  
  {
```

```

    "_id": "6071ad8d4f29c9d6f5393309",
    "content": "Help me, please~",
    "send_time": "2021-04-04T14:25:57.075Z",
    "arrive_time": "2021-04-04T16:25:57.075Z",
    "sender": {
      "user_id": "6071ad8d4f29c9d6f5393309",
      "name": "isabella",
      "country": "Korea",
      "favorites": [
        "Movie"
      ],
      "languages": [
        {
          "name": "ja",
          "level": 0
        }
      ]
    },
    "post_animal": {
      "animal_url": "https://s3.domain/filename.png",
      "head_url": "https://s3.domain/filename.png",
      "top_url": "https://s3.domain/filename.png",
      "pants_url": "https://s3.domain/filename.png",
      "shoes_url": "https://s3.domain/filename.png",
      "gloves_url": "https://s3.domain/filename.png"
    }
  }
]

```

- Fail

HTTP status code : 400 Bad Request

```

{
  "status": 400,
  "message": "해당 mailbox 의 사용자가 아닙니다."
}

```

name	type	description
status	number	HTTP status code(에러 상황에 따라 변할 수 있습니다.)
message	string	에러 메시지(메시지 내용은 에러 상황에 따라 변할 수 있습니다.)

- Leave specific mailbox

Method: GET

URL: /mailboxes/leave/:mailboxId

URL Parameters

name	type	description	example
mailboxId	string	메일함의 id	60681268ab645e32c4191cad

Authorizations : JWT

Request body : None

Response

- Success

HTTP status code : 200 OK

Request Body 가 없습니다. 해당 요청이 완료되면 해당 mailbox 에서 유저 정보가 제외됩니다.

- Fail

HTTP status code : 400 Bad Request

```
{
  "status": 400,
  "message": "해당 mailbox 의 사용자가 아닙니다."
}
```

HTTP status code : 404 Not Found

```
{
  "status": 404,
  "message": "해당 mailbox 는 존재하지 않습니다."
}
```

name	type	description
status	number	HTTP status code(에러 상황에 따라 변할 수 있습니다.)
message	string	에러 메시지(메시지 내용은 에러 상황에 따라 변할 수

있습니다.)

결론

Github repository

- Application: <https://github.com/capstone-design-2021-1-7/Anipal-iOS>
- Server: <https://github.com/capstone-design-2021-1-7/Anipal-Server>

기대 효과

느린 편지라는 키워드를 구체적으로 잡고 진행한 프로젝트인 만큼 기획 의도를 벗어나지 않고 꾸준히 개발하여 향후 마켓 출시까지 무사히 마친다면 안정적인 수요층이 생길 것으로 예상됩니다. 안정감을 주는 콘텐츠와 더불어 동물 캐릭터 커스터마이징이라는 게임적 요소는 이용자들에게 감성과 힐링을 찾아줄 것으로 기대됩니다.

Git 을 이용한 협업에서 더 나아가 애자일 방법론을 적용함으로써 현업에서 실제로 어떻게 스프린트를 나누어 프로젝트를 관리하는지를 알 수 있었으며, 또한 그에 맞는 여러가지 관리 소프트웨어(Trello, Slack 등)를 직접 경험해볼 수 있는 기회가 되었습니다.

아쉬웠던 점

리팩토링을 통해 조금 더 효율적으로 만들 수 있는 부분들이 있었지만 정해진 스케줄을 맞추려다 보니 깊은 고민을 할 수 있는 시간이 없어서 신경 쓰지 못한부분들이 있는 게 아쉬웠습니다.