프론트엔드 API 연동 완전 가이드

📂 파일 구조

```
ra-frontend/
├── js/
│   └── api.js          ← API 클라이언트 (이미 생성됨)
├── index.html
├── login.html
├── signup.html
├── news.html           ← API 연동 필요
└── community.html      ← API 연동 필요
```

🔧 1. news.html에 API 연동

news.html의 `</body>` 태그 직전에 추가:

```html
```

```
<script src="js/api.js"></script>
<script>
    // 페이지 로드 시 뉴스 로드
    let currentPage = 1;
    let currentCategory = '전체';

    async function loadNews(page = 1, category = '전체') {
        try {
            const result = await api.get(
                `/news/list?page=${page}&limit=10&category=${encodeURIComponent(category)}`
            );

            if (result.success) {
                displayNews(result.data.news);
                displayPagination(result.data.pagination);
                currentPage = page;
                currentCategory = category;
            }
        } catch (error) {
            console.error('뉴스 로딩 오류:', error);
            alert('뉴스를 불러오는데 실패했습니다.');
        }
    }

    function displayNews(newsList) {
        const newsContainer = document.querySelector('.news-list');

        if (newsList.length === 0) {
            newsContainer.innerHTML = '<p style="text-align: center; color: #888;">게시글이 없습니다.</p>';
            return;
        }

        newsContainer.innerHTML = newsList.map(news => `
            <article class="news-item">
                <div class="news-thumbnail">
                    ${news.thumbnail ?
                        `<img src="${news.thumbnail}" alt="${news.title}">` :
                        '<div style="font-size: 80px;">📰</div>'
                    }
                </div>
                <div class="news-details">
                    <div class="news-meta">
                        <span class="news-category">${news.category}</span>
                        <span class="news-date">📅 ${formatDate(news.date)}</span>
                    </div>
                    <h3>${news.title}</h3>
```

```
                <p class="news-excerpt">${news.excerpt}</p>
                <a href="news-detail.html?id=${news.id}" class="read-more">
                    자세히 보기 →
                </a>
            </div>
        </article>
    `).join('');
}

function displayPagination(pagination) {
    const paginationContainer = document.querySelector('.pagination');

    let html = '';

    // 이전 버튼
    html += `<button class="page-btn" ${!pagination.hasPrev ? 'disabled' : ''}
            onclick="loadNews(${currentPage - 1}, '${currentCategory}')"><←</button>`;

    // 페이지 번호들
    for (let i = 1; i <= pagination.totalPages; i++) {
        if (i === pagination.currentPage) {
            html += `<button class="page-btn active">${i}</button>`;
        } else {
            html += `<button class="page-btn" onclick="loadNews(${i}, '${currentCategory}')">${i}</button>`;
        }
    }

    // 다음 버튼
    html += `<button class="page-btn" ${!pagination.hasNext ? 'disabled' : ''}
            onclick="loadNews(${currentPage + 1}, '${currentCategory}')">→</button>`;

    paginationContainer.innerHTML = html;
}

function formatDate(dateString) {
    const date = new Date(dateString);
    const year = date.getFullYear();
    const month = String(date.getMonth() + 1).padStart(2, '0');
    const day = String(date.getDate()).padStart(2, '0');
    return `${year}.${month}.${day}`;
}

// 카테고리 필터 클릭 이벤트
document.querySelectorAll('.filter-tab').forEach(tab => {
    tab.addEventListener('click', function() {
        document.querySelectorAll('.filter-tab').forEach(t => t.classList.remove('active'));
        this.classList.add('active');
```

```
            const category = this.textContent.trim();
            loadNews(1, category);
        });
    });

    // 페이지 로드 시 첫 뉴스 로드
    window.addEventListener('DOMContentLoaded', () => {
        loadNews(1, '전체');
    });
</script>
```

🔧 2. community.html에 API 연동

community.html의 `</body>` 태그 직전에 추가:

```
html
```

```
<script src="js/api.js"></script>
<script>
    let currentPage = 1;
    let currentCategory = '전체';

    // 게시글 로드
    async function loadPosts(category = '전체', page = 1) {
        try {
            const result = await api.get(
                `/community/posts?page=${page}&limit=20&category=${encodeURIComponent(category)}&sort=l
            );

            if (result.success) {
                displayPosts(result.data.posts);
                displayPagination(result.data.pagination);
                currentPage = page;
                currentCategory = category;
            }
        } catch (error) {
            console.error('게시글 로딩 오류:', error);
            alert('게시글을 불러오는데 실패했습니다.');
        }
    }

    function displayPosts(posts) {
        const postsContainer = document.querySelector('.main-content');
        const searchBar = postsContainer.querySelector('.search-bar');
        const pagination = postsContainer.querySelector('.pagination');

        // 기존 게시글 제거 (검색바와 페이지네이션은 유지)
        const existingPosts = postsContainer.querySelectorAll('.post-card');
        existingPosts.forEach(post => post.remove());

        if (posts.length === 0) {
            const noPostsMsg = document.createElement('p');
            noPostsMsg.style.cssText = 'text-align: center; color: #888; padding: 40px;';
            noPostsMsg.textContent = '게시글이 없습니다.';
            postsContainer.insertBefore(noPostsMsg, pagination);
            return;
        }

        // 게시글 추가
        posts.forEach(post => {
            const postCard = createPostCard(post);
            postsContainer.insertBefore(postCard, pagination);
        });
```

```
}

function createPostCard(post) {
    const article = document.createElement('article');
    article.className = 'post-card';

    article.innerHTML = `
        <div class="post-header">
            <div class="post-info">
                <h3 class="post-title">
                    ${post.title}
                    ${post.isHot ? '<span class="hot-badge">HOT</span>' : ''}
                </h3>
                <div class="post-meta">
                    <div class="post-author">
                        <div class="author-avatar">${post.author.username[0]}</div>
                        <span>${post.author.username}</span>
                    </div>
                    <span>📅 ${formatTimeAgo(post.createdAt)}</span>
                </div>
            </div>
            <div class="post-tags">
                <span class="post-tag">${post.category}</span>
            </div>
        </div>
        <p class="post-excerpt">${post.excerpt}</p>
        <div class="post-stats">
            <div class="stat-item">
                <span class="stat-icon">👁</span>
                <span>${post.stats.views}</span>
            </div>
            <div class="stat-item">
                <span class="stat-icon">👍</span>
                <span>${post.stats.likes}</span>
            </div>
            <div class="stat-item">
                <span class="stat-icon">💬</span>
                <span>${post.stats.comments}</span>
            </div>
        </div>
    `;

    // 클릭 시 상세 페이지로 이동
    article.style.cursor = 'pointer';
    article.addEventListener('click', () => {
        window.location.href = `post-detail.html?id=${post.id}`;
    });
```

```
    return article;
}

function displayPagination(pagination) {
    const paginationContainer = document.querySelector('.pagination');

    let html = '';

    html += `<button class="page-btn" ${pagination.currentPage === 1 ? 'disabled' : ''}
            onclick="loadPosts('${currentCategory}', ${currentPage - 1})"><←</button>`;

    for (let i = 1; i <= pagination.totalPages; i++) {
        if (i === pagination.currentPage) {
            html += `<button class="page-btn active">${i}</button>`;
        } else {
            html += `<button class="page-btn" onclick="loadPosts('${currentCategory}', ${i})">${i}</button>`;
        }
    }

    html += `<button class="page-btn" ${pagination.currentPage === pagination.totalPages ? 'disabled' : ''}
            onclick="loadPosts('${currentCategory}', ${currentPage + 1})">→</button>`;

    paginationContainer.innerHTML = html;
}

function formatTimeAgo(dateString) {
    const date = new Date(dateString);
    const now = new Date();
    const diffMs = now - date;
    const diffHours = Math.floor(diffMs / (1000 * 60 * 60));
    const diffDays = Math.floor(diffMs / (1000 * 60 * 60 * 24));

    if (diffHours < 1) return '방금 전';
    if (diffHours < 24) return `${diffHours}시간 전`;
    if (diffDays < 7) return `${diffDays}일 전`;

    return date.toLocaleDateString('ko-KR');
}

// 카테고리 클릭
document.querySelectorAll('.sidebar-category').forEach(category => {
    category.addEventListener('click', function() {
        document.querySelectorAll('.sidebar-category').forEach(c => c.classList.remove('active'));
        this.classList.add('active');

        const categoryName = this.textContent.trim();
```

```
            loadPosts(categoryName, 1);
        });
    });

    // 글쓰기 버튼
    document.querySelector('.create-post-btn').addEventListener('click', function() {
        const token = localStorage.getItem('accessToken');
        if (!token) {
            alert('로그인이 필요한 기능입니다.');
            window.location.href = 'login.html';
        } else {
            // 글쓰기 페이지로 이동 (추후 구현)
            alert('글쓰기 기능은 준비 중입니다.');
        }
    });

    // 검색 기능
    document.querySelector('.search-btn').addEventListener('click', async function() {
        const searchValue = document.querySelector('.search-input').value.trim();
        if (!searchValue) {
            alert('검색어를 입력해주세요.');
            return;
        }

        try {
            const result = await api.get(`/community/search?q=${encodeURIComponent(searchValue)}&page=1`);
            if (result.success) {
                displayPosts(result.data.posts);
                displayPagination(result.data.pagination);
            }
        } catch (error) {
            console.error('검색 오류:', error);
            alert('검색 중 오류가 발생했습니다.');
        }
    });

    // Enter 키로 검색
    document.querySelector('.search-input').addEventListener('keypress', function(e) {
        if (e.key === 'Enter') {
            document.querySelector('.search-btn').click();
        }
    });

    // 페이지 로드 시
    window.addEventListener('DOMContentLoaded', () => {
        loadPosts('전체', 1);
```

```
    });
</script>
```

🔧 3. index.html에 로그인 상태 표시

index.html의 `</body>` 태그 직전에 추가:

```html
<script src="js/api.js"></script>
<script>
    // 로그인 상태 확인 및 UI 업데이트
    window.addEventListener('DOMContentLoaded', () => {
        const user = JSON.parse(localStorage.getItem('user') || 'null');
        const navLinks = document.querySelector('.nav-links');

        if (user) {
            // 로그인된 경우 - 로그인 링크를 사용자 정보로 변경
            const loginLink = navLinks.querySelector('a[href="login.html"]');
            if (loginLink) {
                const li = loginLink.parentElement;
                li.innerHTML = `
                    <a href="#" style="color: #e94560;" onclick="handleLogout(event)">
                        ${user.username} (로그아웃)
                    </a>
                `;
            }
        }
    });

    // 로그아웃 처리
    function handleLogout(event) {
        event.preventDefault();

        if (confirm('로그아웃 하시겠습니까?')) {
            // 토큰 삭제
            localStorage.removeItem('accessToken');
            localStorage.removeItem('refreshToken');
            localStorage.removeItem('user');

            alert('로그아웃 되었습니다.');
            window.location.reload();
        }
    }
</script>
```

✅ 4. 테스트 체크리스트

백엔드 테스트

```bash
# 1. 백엔드 실행
cd ra-backend
npm run dev

# 2. 터미널에서 다음 메시지 확인:
# 🚀 Server running on http://localhost:3000
# 📱 Environment: development
```

API 직접 테스트 (브라우저 콘솔 또는 Postman)

1. 회원가입 테스트:

```javascript
fetch('http://localhost:3000/api/auth/signup', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    username: 'testuser',
    email: 'test@example.com',
    password: 'Test1234!',
    birthdate: '1990-01-01',
    region: 'kr',
    marketing: true
  })
}).then(r => r.json()).then(console.log);
```

2. 로그인 테스트:

```javascript
```

```
fetch('http://localhost:3000/api/auth/login', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({
        email: 'test@example.com',
        password: 'Test1234!',
        remember: true
    })
}).then(r => r.json()).then(console.log);
```

3. 뉴스 목록 테스트:

```
javascript

fetch('http://localhost:3000/api/news/list?page=1&limit=10&category=전체')
    .then(r => r.json())
    .then(console.log);
```

4. 커뮤니티 게시글 테스트:

```
javascript

fetch('http://localhost:3000/api/community/posts?page=1&limit=20&category=전체')
    .then(r => r.json())
    .then(console.log);
```

프론트엔드 테스트

```
bash

# 프론트엔드 폴더에서 실행
cd ra-frontend
python -m http.server 8080
```

브라우저에서 테스트:

1. http://localhost:8080 접속
2. 회원가입 페이지에서 계정 생성
3. 로그인 페이지에서 로그인
4. 개발자 도구(F12) → Console에서 토큰 확인:

```
javascript

    localStorage.getItem('accessToken')
```

5. news.html 접속 → 뉴스 목록 확인

6. community.html 접속 → 게시글 목록 확인

🐛 문제 해결

문제 1: CORS 에러

```
Access to fetch at 'http://localhost:3000' has been blocked by CORS policy
```

해결: 백엔드 `.env` 파일 확인

```
env

ALLOWED_ORIGINS=http://localhost:8080,http://127.0.0.1:8080
```

문제 2: 뉴스/게시글이 안 보임

해결: 데이터베이스에 테스트 데이터 추가

```
sql

INSERT INTO news (title, category, excerpt, content, author)
VALUES ('테스트 뉴스', '공지사항', '테스트 내용', '상세 내용', 'Admin');
```

문제 3: 로그인 후 토큰이 저장 안 됨

해결: 브라우저 콘솔에서 확인

```
javascript

// 로그인 응답 확인
console.log(result);

// 토큰 저장 확인
console.log(localStorage.getItem('accessToken'));
```

📊 완성도 체크

- ✅ 백엔드 서버 실행
- ✅ 데이터베이스 연결
- ✅ 회원가입 API 동작
- ✅ 로그인 API 동작

- ✅ 토큰 저장 및 인증
- ✅ 뉴스 API 동작
- ✅ 커뮤니티 API 동작
- ✅ 프론트엔드 연동 완료

모든 체크가 완료되면 기본적인 백엔드-프론트엔드 연동이 완성됩니다! 🎉