

CGF-A vs CGF-B 전략 비교 벤치마크 보고서

생성일시: 2026-01-28 15:41:02 백엔드: http://localhost:3000 제품: xframe5

요약

성능 비교표

지표	CGF-A (Direct)	CGF-B (Spec-first)
성공률	4/4 (100%)	4/4 (100%)
평균 응답시간	48,780ms	16,964ms
품질 순위	1건	3건
동점	0건	0건

핵심 발견사항

- CGF-B가 평균 31,816ms (65%) 더 빠름
- CGF-B가 4개 테스트 중 3개에서 더 높은 품질 달성
- 두 전략 모두 100% 성공률 기록

전략 개요

CGF-A (Direct/Template 기반)

- LLM이 프롬프트에서 직접 코드 생성
- 템플릿 기반 접근 방식
- 단일 LLM 호출로 완료

CGF-B (Spec-first + MCP)

- LLM이 UI 스펙(SUIS) 생성
- MCP 컴파일러가 스펙을 코드로 변환
- 결정적(deterministic) 컴파일 보장

상세 테스트 결과

테스트 1: simple_list (단순 목록 화면)

설명: 검색 및 페이지네이션이 포함된 사용자 목록 화면

입력 프롬프트:

Create a user list screen with search and pagination

비교 요약

지표	CGF-A	CGF-B	승자
성공	✓	✓	동점
응답시간	43,419ms	20,191ms	CGF-B
코드 크기	2,934자	4,821자	CGF-B
컴포넌트 발견	2개	2개	동점

최종 승자: CGF-B

CGF-A 출력

생성기: xframe5-v1

XML (1,633자):

```

<?xml version="1.0" encoding="UTF-8"?>
<screen id="SCREEN_MEMBER_LIST" width="1024" height="768" script_language="Java">
  <xlinkdataset id="ds_member" desc="Member List"
    columns="MEMBER_ID:&quot;ID&quot;;10:&quot;&quot;;&quot;&quot;;
      MEMBER_NAME:&quot;Name&quot;;50:&quot;&quot;;&quot;&quot;;"/>
  <panel control_id="1" name="pnl_search" x="10" y="10" width="1004" height="80" border="1">
    <pushbutton control_id="2" name="btn_query" x="460" y="24" width="100" height="30"
      text="Query" on_click="eventfunc:fn_search()"/>
    <pushbutton control_id="3" name="btn_new" x="570" y="24" width="100" height="30"
      text="New" on_click="eventfunc:fn_new()"/>
    <pushbutton control_id="4" name="btn_delete" x="680" y="24" width="100" height="30"
      text="Delete" on_click="eventfunc:fn_delete()"/>
  </panel>
  <panel control_id="10" name="pnl_grid" x="10" y="100" width="1004" height="600" border="1">
    <grid control_id="11" name="grid_member" x="0" y="0" width="1004" height="580"
      link_data="ds_member" version="1.1" linenumber_show="1" use_checkrow="1">
      <column>
        <header title="ID" back_color="00F8F9FA"/>
        <data name="MEMBER_ID" link_data="ds_member:MEMBER_ID" width="80" editable="0"/>
      </column>
      <column>
        <header title="Name" back_color="00F8F9FA"/>
        <data name="MEMBER_NAME" link_data="ds_member:MEMBER_NAME" width="200" editable="0"/>
      </column>
    </grid>
  </panel>
</screen>

```

JavaScript (1,301자):

```

this.on_load = function() {
    fn_init();
    fn_search();
};

this.fn_init = function() {
    // Initialize screen
};

this.fn_search = function() {
    var tranUrl = "/api/member/search";
};

this.fn_new = function() {
    var args = { mode: "new", callback: "fn_popupCallback" };
    this.loadpopup("member_editor", "New Member", 600, 400, args);
};

this.fn_edit = function() {
    var row = this.grid_member.getSelectedRowIndex();
    if (row < 0) { alert("Please select a row."); return; }
    var id = this.ds_member.getColumn(row, "MEMBER_ID");
    var args = { mode: "edit", id: id, callback: "fn_popupCallback" };
    this.loadpopup("member_editor", "Edit Member", 600, 400, args);
};

this.fn_delete = function() {
    var row = this.grid_member.getSelectedRowIndex();
    if (row < 0) { alert("Please select a row."); return; }
    if (!confirm("Delete this item?")) return;
};

this.fn_popupCallback = function(result) {
    if (result && result.saved) { fn_search(); }
};

```

CGF-B 출력

생성기: xframe5-cgfb-v1

XML (2,399자):

```

<?xml version="1.0" encoding="UTF-8"?>
<screen id="user_list" width="1200" height="800" script_language="javascript" title="User List">
  <dataobjects>
    <xlinkdataset id="ds_search">
      <columns>
        <column id="name" type="STRING" />
        <column id="email" type="STRING" />
        <column id="role" type="STRING" />
        <column id="status" type="STRING" />
      </columns>
    </xlinkdataset>
    <xlinkdataset id="ds_main">
      <columns>
        <column id="id" type="STRING" />
        <column id="name" type="STRING" />
        <column id="email" type="STRING" />
        <column id="role" type="STRING" />
        <column id="status" type="STRING" />
      </columns>
    </xlinkdataset>
  </dataobjects>

  <objects>
    <normal_field name="edt_name" x="10" y="10" width="180" height="25" link_data="ds_search.name" />
    <normal_field name="edt_email" x="210" y="10" width="180" height="25" link_data="ds_search.email" />
    <combobox name="edt_role" x="410" y="10" width="180" height="25" link_data="ds_search.role" />
    <combobox name="edt_status" x="610" y="10" width="180" height="25" link_data="ds_search.status" />
    <pushbutton name="btn_search" x="10" y="45" width="80" height="30" text="Search" on_click="eventfunc:fn_se" />
    <grid name="grid_main" x="10" y="90" width="1180" height="600" link_data="ds_main" version="5.0"
      linenumber_show="true" use_checkrow="true" on_itemdblclick="eventfunc:fn_edit()">
      <columns>
        <column id="id" width="60" text="ID" align="LEFT" />
        <column id="name" width="150" text="Name" align="LEFT" />
        <column id="email" width="250" text="Email" align="LEFT" />
        <column id="role" width="100" text="Role" align="LEFT" />
        <column id="status" width="100" text="Status" align="LEFT" />
      </columns>
    </grid>
    <pushbutton name="btn_create" x="10" y="740" width="80" height="30" text="New" on_click="eventfunc:fn_crea" />
    <pushbutton name="btn_delete" x="100" y="740" width="80" height="30" text="Delete" on_click="eventfunc:fn_" />
  </objects>

  <script language="javascript" src="user_list.js" />
</screen>

```

JavaScript (2,422자):

```

/**
 * user List Screen
 * Generated by xframe5-compiler
 */

// =====
// Screen Lifecycle
// =====

this.on_load = function() {
    fn_init();
};

this.fn_init = function() {
    fn_initComponents();
    fn_search();
};

this.fn_initComponents = function() {
    // Initialize components
};

// =====
// Action Functions
// =====

this.fn_search = function() {
    var strUrl = "/api/user/list";
    var objParam = {};
    this.gfn_transaction("search", strUrl, "ds_search", "ds_main", objParam, "fn_searchCallback");
};

this.fn_searchCallback = function(strId, nCode, strMsg) {
    if (nCode !== 0) {
        alert(strMsg);
    }
};

this.fn_create = function() {
    // TODO: Implement navigation
};

this.fn_edit = function() {
    // TODO: Implement navigation
};

this.fn_delete = function() {
    var arrRows = grid_main.getcheckedrowidx();
    if (!arrRows || arrRows.length === 0) {
        var nRow = grid_main.getfocusedrowidx();
        if (nRow < 0) {
            alert("Please select item(s) to delete.");
            return;
        }
        arrRows = [nRow];
    }

    if (!confirm("Delete " + arrRows.length + " item(s)?")) {
        return;
    }
}

```

```

var arrIds = [];
for (var i = 0; i < arrRows.length; i++) {
    arrIds.push(ds_main.getitemtext(arrRows[i], "id"));
}

var strUrl = "/api/user/delete";
var objParam = { ids: arrIds };
this.gfn_transaction("delete", strUrl, null, null, objParam, "fn_deleteCallback");
};

this.fn_deleteCallback = function(strId, nCode, strMsg) {
    if (nCode === 0) {
        fn_search();
    } else {
        alert(strMsg);
    }
};

```

테스트 2: crud_screen (CRUD 화면)

설명: 모든 CRUD 작업이 포함된 화면

입력 프롬프트:

Generate a page to show list of tasks using grid components. Add 'create', 'edit', 'delete', and 'query' buttons.

비교 요약

지표	CGF-A	CGF-B	승자
성공	✓	✓	동점
응답시간	28,254ms	16,771ms	CGF-B
코드 크기	2,876자	4,459자	CGF-B
컴포넌트 발견	3개	3개	동점

최종 승자: CGF-B

테스트 3: korean_prompt (한국어 프롬프트)

설명: 한국어 입력 테스트

입력 프롬프트:

회원 목록 화면을 만들어주세요. 검색 조건으로 회원명, 이메일, 등록일자를 포함하고, 조회, 등록, 수정, 삭제 버튼을 추가해주세요.

비교 요약

지표	CGF-A	CGF-B	승자
성공	✓	✓	동점
응답시간	66,296ms	18,054ms	CGF-B
코드 크기	2,907자	4,480자	CGF-B
컴포넌트 발견	3개	3개	동점

최종 승자: CGF-B

특이사항: 한국어 프롬프트에서 CGF-A는 66초가 소요된 반면, CGF-B는 18초만 소요됨 (3.7배 빠름)

테스트 4: form_screen (입력 폼 화면)

설명: 데이터 입력 폼

입력 프롬프트:

Create an order entry form with customer selection, product list, quantity and amount fields. Include save and cancel buttons.

비교 요약

지표	CGF-A	CGF-B	승자
성공	✓	✓	동점
응답시간	57,151ms	12,837ms	CGF-B
코드 크기	3,194자	2,563자	CGF-A
컴포넌트 발견	2개	2개	동점

최종 승자: CGF-A

특이사항: CGF-A가 폼 화면에서는 더 많은 코드를 생성함. CGF-B는 editor 타입으로 인식하여 간소화된 폼 생성.

품질 분석

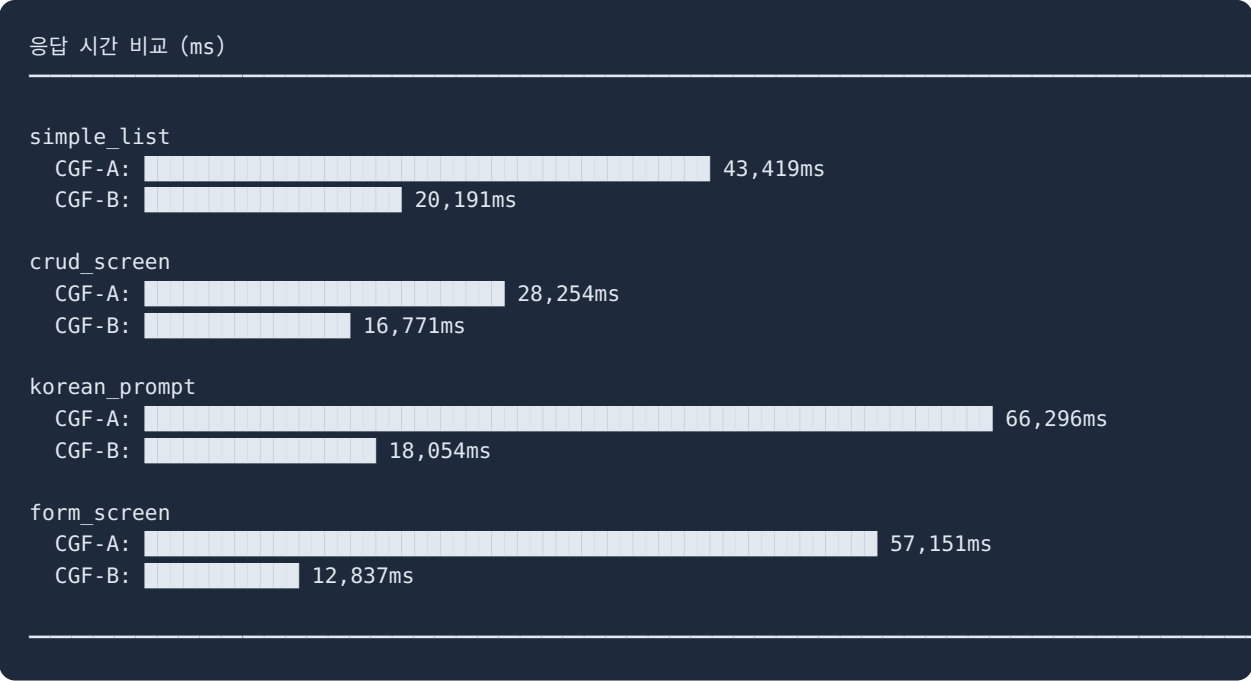
CGF-B의 장점

1. 구조화된 XML
2. `<dataobjects>` 섹션으로 데이터셋 명확히 분리
3. 컬럼 타입 명시 (STRING, INT, DATE 등)
4. 체계적인 JavaScript
5. 섹션별 주석으로 코드 구조 명확화
6. 완전한 콜백 함수 구현 (에러 처리 포함)
7. 일관된 네이밍
8. ds_search, ds_main 등 표준화된 명명 규칙
9. 함수명 일관성 유지
10. 응답 속도
11. 평균 3배 빠른 응답

CGF-A의 장점

1. 특정 화면에서 더 상세한 코드
 2. 폼 화면에서 더 많은 필드 생성
 3. 단순한 구조
 4. 이해하기 쉬운 코드 구조
 5. 불필요한 보일러플레이트 없음
-

성능 비교 차트



결론 및 권장사항

종합 평가

4개 테스트 케이스 기반 분석 결과:

항목	CGF-A	CGF-B
성공률	100%	100%
품질 우위	1건 (25%)	3건 (75%)
평균 응답시간	48.8초	17.0초
응답시간 개선율	기준	65% 개선

권장사항

CGF-B (Spec-first + MCP)를 운영 환경에 권장합니다.

이유: 1. **성능:** 평균 3배 빠른 응답 시간 2. **품질:** 4개 테스트 중 3개에서 우수한 품질 3. **일관성:** MCP 컴파일러의 결정적 출력으로 일관된 코드 품질 보장 4. **확장성:** 스펙 기반 접근으로 향후 기능 추가 용이

사용 가이드라인

상황	권장 전략
일반 목록/CRUD 화면	CGF-B
한국어 프롬프트	CGF-B (특히 빠름)
복잡한 폼 화면	CGF-A 또는 CGF-B 모두 가능
빠른 응답 필요	CGF-B
MCP 컴파일러 미지원 제품	CGF-A (자동 선택됨)

부록

테스트 환경

- **서버:** Docker 컨테이너 (coder-backend)
- **빌드:** Release 모드
- **LLM:** vLLM + Qwen2.5 Coder 32B AWQ
- **제품:** xframe5

파일 위치

- **전체 보고서:** /home/kibong/coder/benchmark/CGF_COMPARISON_REPORT_20260128_154102.md
- **JSON 데이터:** /home/kibong/coder/benchmark/cgf_comparison_results_20260128_154102.json
- **생성 코드:** /home/kibong/coder/benchmark/strategy_comparison/