



# Opportunities Exchange

## FROM SILOS TO SYSTEMS: Unlocking Child Care Data with APIs

By Adam Lucas and Louise Stoney

July, 2025



State child care administrators, as well as the families and providers they serve, spend many hours managing data—filling out forms, entering information into myriad databases, verifying documentation, double-checking entries to prevent errors, conducting desk audits to prevent fraud, and countless other routine tasks. Modern technology has capacity to not only streamline many of these tasks but make sure that information is correct the first time it is collected.

The key to enabling timely and accurate data collection is building data bridges—which technologists call Application Programming Interfaces, or APIs. We are all familiar with this approach. It's how the platforms many of us use every day—such as Facebook, Travelocity, Zillow, electronic banking apps, among others—share data.

A growing number of states are starting to build APIs designed to support the process of managing child care financial assistance, program licensing, quality rating, and more. States on the cusp of innovation are going even further—using APIs to link siloed systems and enable statewide, cross-sector data collection and analysis.

While the approach holds great promise, the devil is in the details. All too often well-intentioned states craft APIs that are not widely used or do not work as intended. To help address this challenge, [Opportunities Exchange](#) (OppEx) took a deep dive into the child care API process to identify best practice. This brief summarizes lessons learned and is intended to serve as a guide for states seeking to harness the power of modern technology for the purpose of streamlining child care subsidy administration, program reporting, monitoring and other routine tasks.

### KEY LESSONS

While each state is unique, our research underscored six cross-cutting issues that must be addressed when planning and launching an API. These include:

#### Commitment

Implementing a successful API requires strong leadership focused on ensuring that all stakeholders understand the goal and are willing and able to support change. This will take time, and likely require multiple conversations, technology demonstrations, opportunities to ask questions and share concerns. A common challenge for many stakeholders is that an API is *based on collecting data rather than filling out forms*. State staff are accustomed to tracking compliance based on completion of specific forms, but when data are collection via an API those forms may not be needed at all. Similarly, staff who are accustomed to working with paper documents might not trust that data held in an electronic data base are valid or accurate. If



# Opportunities Exchange

## Indiana

The Indiana Office of Early Childhood and Out-of-School Learning (OECOSL) launched a [web page](#) to feature CCMS Frequently Asked Questions. A centralized location for information sharing makes it possible to continually identify, and respond to, new questions so that information is current and learning is shared quickly.

these concerns are not heard, and addressed throughout the planning process, even the best API will not be effectively used.

### Communication

Communication is key to nurturing and sustaining commitment. Everyone involved in the change needs to understand the process, the timeline, and progress to date. Regular updates, linked to documents that repeat (over and over) the goal and desired timeline, should be shared with everyone that has a role to play. When information is lacking, gossip and rumors grow to fill in the blanks, and misinformation can kill even the best projects. In short, it is impossible to communicate too often.

Communicating in various formats is helpful, including one-page summaries, Frequently Asked Questions, a dedicated website, a hotline, presentations at regional conferences, webinars and more.

### Detailed Workflow Maps

Effectively crafting and implementing an API requires a deep understanding of every step in the data journey. (The devil is always in the details!) Thus, the most important first step in planning an API is mapping both current practice and the desired future practice. This means walking through, and documenting, every step in the process for every end user. Crafting a workflow map typically involves pretending to be a parent/provider/coach/administrator, walking through each step these end users must take, including reviewing each form they must complete and system into which they must enter data. Interviews are an important way to collect this information and help to identify the processes that are most challenging and the 'edge cases' that can lead to errors or delays.

If a step is required in the current workflow, it *must* be accounted for in the map. Information for the *current* workflow map will likely document data collected via the many forms that parents, providers and state staff must complete or verify. The *desired* workflow map, on the other hand, will focus on identifying how, and where, these data could be obtained without requiring a specific form. (For example, rather than verifying attendance by requiring parents to sign, and providers to submit, daily check-in/out forms, an API could pull data from electronic check-in via Child Care Management Software or CCMS.) When crafting workflow maps, make sure to map as many potential 'edge cases' as possible. Typically edge cases are found in processes such as:

- Allowing providers to submit records for historical pay periods.  
(How far back are providers allowed to bill?)
- Allowing children to attend multiple providers during a pay period.  
(How does the state document and calculate payment in these cases?)
- Determining part-time vs full-time care.  
(How does the state determine the correct payment category?)
- Handling situations when a parent switches to a completely new provider during a billing cycle.  
(How does each provider bill for a partial week?)

### Testing

If anything can go wrong, it will. Testing the API, multiple times with various end users, is essential. While tests can involve a small number of cases, to effectively identify problems tests should use real data (such as the edge cases described above or historical data) as well as multiple end users and a range of provider types and technology vendors. All too often, testing is done with artificially generated data created with the end goal in mind, rather than data that reflects real-world experience. If real-world data are used, even if it is historical, testing is more likely to reveal potential problems, including additional edge cases that need to be brought back to the workflow mapping. Ensuring that the workflow mapping team is aware of the many situations where data can fail to transfer, and why, is key to ensuring successful API development.



### Training

A training component, focused on ensuring that all stakeholders understand how to use the API, should be crafted and tested as well. Everyone that is expected to use the data bridge must be aware of the change, understand their role, and know where to go if they have questions or concerns. In addition to providers and families, this may include training for industry intermediaries such as Child Care Resource and Referral Agencies, Training and Technical Assistance Partners, provider associations, and others.

Training will likely include how to access and share information via the API (vs signing into a provider portal on the state system). However, best practice is to ensure that training is also designed to help end users and support staff learn about the myriad edge cases that can lead to an API failure as well as what to do when problems arise. Remember—if anything can go wrong, it will. If key informants are aware of potential pitfalls, and prepared to promptly address them, roll-out will be smoother and trust can be maintained even when circumstances are challenging.

### Pilot or Phased Launch

All too often states feel pressured to rush into a statewide launch. If possible, this should be avoided. It often takes only one mistake to fuel distrust, and once a large statewide launch is in action it is hard to back-pedal. If the launch occurs in phases, then glitches can be worked out along the way as new end users are brought on board.

Moreover, the phased launch should assume full and accurate submission of data for all providers that elect to participate in the Phase I roll-out. Allowing providers a “grace period” (e.g. they can choose to go back to paper billing) or flexibility in submissions (e.g. accepting partial participation) can lead to greater confusion and slower uptake. Providers who are not ready to fully embrace technology are not good candidates for a Phase I pilot.

### APIs IN ACTION: COMMON CHALLENGES

The lessons described above are guiding principles for API development and launch. However effective implementation requires close attention to details as there are myriad reasons why an API can go awry. A common challenge in many child care APIs is inability to accurately link the Provider ID or the Child ID. In situations where a state seeks to use an API to track attendance, failure often occurs when the child’s classroom schedule does not align or state policy regarding absences requires complex coding. The tables below outline steps to address each of these challenges.

### Setting up Integration Inside a CCMS (*Linking Provider ID*)

Provider Role	State Role	Training/Technical Assistance Role	API Function	Reason
Provider must “link” their attendance data to the state’s database by entering their Provider ID into the CCMS they use for billing. (This may, or may not, be the provider’s License ID.)	The state must make access to the Provider ID available in their system, along with a method for the provider to find and link to this ID.	The T/TA entity must make sure the provider knows how to: 1) find their Provider ID; 2) format the ID appropriately; and, 3) enter the ID in the appropriate place in each participating CCMS.	The API will use the ID submitted by the provider to validate the submission against all Provider IDs currently active in the state system. An error message should be sent if the ID is not found or not active.	The ID is the main provider record in the state system where all relevant data will be held. Any information on attendance, children/families, licensing, staffing, etc. will be associated with a provider ID and can be validated.

### Challenges to Consider:

- Often states have multiple technology systems that providers must use, which can make implementation of an API (that may apply to only one system) confusing at first. Clear communication, which directs providers to the correct system, and the correct Provider and Child ID, is essential. If the state does not use the provider’s License ID, this step is even more important.
- Best practice for implementing a new provider policy/process is to NOT grant exceptions or a grace period. The new process should be a requirement, and providers should be held to the new process from day one. If you allow for mistakes,



or grant a 1-month grace period, it will likely lead to providers not understanding the new process and assume that how submission was made during the grace period was correct. (This is yet another reason for launching APIs in phases or starting with a small pilot. Communicate clearly and hold firm to the change from the outset. Then test to make sure it works before scaling.)

Connecting Children Via the API				
Provider Role	State Role	Training/Technical Assistance Role	API Function	Reason
The provider must “link” all approved children to match the child IDs used by the state subsidy system. Providers should be able to easily find the Child ID.	The state must make available the IDs of all approved children in their system and a method for the provider to find this ID.	The trainer must make sure the provider knows how to: 1) find the Child ID; 2) format the ID appropriately; and, 3) enter the ID in the appropriate place in each participating CCMS.	The API will use the Child ID to validate the submission against IDs currently active in the state system. An error message should be sent if the ID is not found or not active.	The ID is the main child record in the state system where all relevant data will be held. Any attendance or absence information will be associated with a child ID and can be validated. A unique ID is important because matching against name and/or date of birth is not guaranteed to be accurate.

### Challenges to Consider:

- The state must be sure to allow access to all children for valid submission ranges. For example, if a provider is allowed to back-bill for a child who attended during a prior submission period, the system should be validating the data submission against children who were eligible and active in the previous submission period (and not the current one).
- It is important to timely sync all subsidy approvals with the system responsible for the API connection. Delays pushing data from a case management system to the system handling the API will cause error messages for children who are approved and active. If access to the ID is delayed, the provider will be unable to bill for all enrolled children, the submission will likely fail, and the provider will not be paid—causing serious cash flow problems. Payment delays and error messages will quickly diminish use of an API.
- It is important to ensure that a child’s ID remains consistent throughout the subsidy process. If, for example, a new ID is assigned each time subsidy is re-authorized, then a child will have multiple IDs linked to different time periods. While providers can be trained to use multiple child IDs based on the authorization period, this is not best practice as it leads to confusion and will result in API errors.

Submitting Attendance Via the API				
Provider Role	State Role	Training/Technical Assistance Role	API Function	Reason
The pay period selected by the provider within their CCMS MUST align with the pay period used by the state.  Each child must have an expected daily attendance field, as well as check in AND check out times AND/OR a recorded absence (with reason).	The state should ensure that pay periods accurately align with provider billing practices.  The state must clearly communicate how absences are coded. If an absence reason code is required, the provider should be able to enter all allowed absence types directly in the CCMS. Best practice is to have only 2 or 3 codes to document reason for absence.	T/TA entities must fully understand how to set up billing periods in the CCMS to align with state subsidy billing periods and how to code absences. If there are other specific requirements or other potential error messages, T/TA entities must know how to troubleshoot each potential error.	The API will validate/confirm all the child, provider and billing data and respond with any potential errors (examples below).	Attendance submission can be complicated due to strict data requirements. When problems arise, error messages should be specific, non-technical, and actionable. General statements like “an error has occurred” or simply rejecting submissions without explanation will lead to confusion, delays and frustration, and ultimately have a chilling effect on use of the API.



# Opportunities Exchange

## Kentucky

The Kentucky Department of Human Services has been working to harness the power of brightwheel Child Care Management Software to streamline a host of state reporting requirements, including tracking attendance, licensing compliance, gathering supply data, and more. The process has also led to several important changes in state policy, such as reducing the number of absence codes from nine to four. This seemingly minor change will not only reduce errors and save time but will likely increase provider engagement.

### Challenges to Consider:

- If absence reasons/codes are required, they need to be synced with CCMS. Best practice is to work with CCMS vendors and request prompts for absence reason/code during data entry.
- Data designed for the test won't reveal problems. It is important to test attendance tracking via API with *real* data. Using sample data populated by developers will result in missing edge cases and lead to an extended pilot as vendors try to align to real-world experience. In some cases, historical data can be a good indicator of potential data inconsistencies.
- If edge cases crop up during a pilot, implementation should be paused and the workflow analyzed based on end-user interviews. If errors are frequent, and not addressed quickly, providers (and industry intermediaries) will quickly lose faith in the API and the whole project could go south.
- Edge cases offer teachable moments. Consider using examples of what edge cases cropped up to inform change as well as guide training for providers and T/TA entities.
- Vague error messages can be a significant challenge. It is helpful to document all error messages and resolutions to inform change going forward. The document should also identify who "owns" the resolution. The teams responsible for fixing an error should be aware of this document, and the providers should be pointed to the responsible team for each issue. The document can, and should, continue to evolve as issues arise. A detailed list of common errors and validation challenges is included below.
- Inconsistent implementation can underscore problems that must be addressed. Contracting with multiple CCMS during the development of the API will help leaders understand challenges, errors and use patterns. If one vendor API is breaking, but the other two are working, the resolution likely involves a deeper dive into the broken API and work with that specific vendor.
- Funding that is not aligned with goals can have unintended consequences. To this end, it is important to establish funding milestones to match internal goals. For example, a state might begin by offering free CCMS subscriptions to all providers for a limited period and then continue to pay for a free subscription for providers that use the CCMS API to submit attendance.
- To accurately transmit information, an API must validate all fields before sending data. If *any* of the fields produce an error, *nothing* should be stored in a state system. The acceptance of partial submissions creates a major data mismatch and would require a manual submission to fix those unaccepted records.

### COMMON API ERRORS AND VALIDATIONS

Listed below are problems that must typically be accompanied by a detailed error message along with actionable next steps.

- Is the provider ID correct?
- Is the pay period valid?
- Are all children being submitted for payment valid for the selected pay period?
  - ▲ Some children may have attended for only part of the month or week. If there is a submission for only part of the pay period, the state must determine how to handle that information. Options include:
    - Capture and store it all;
    - Capture and store just the correct pay period;
    - Follow your current manual process.
- For EACH child and EACH attendance record, the API must scan to ensure adherence to any restrictions on this data:
  - ▲ Is there a check in AND check out time on every day?
  - ▲ Are the check in and check out times overlapping?
  - ▲ Is there a large gap of unaccounted for time in the middle of the day?
  - ▲ Is there an absence recorded and does the absence contain the appropriate coding? (sick, vacation, provider closed, etc)
- Are there any other rules that would block the submission of the current manual



entry form? Any data anomalies that would trigger actions or questions on the state side of their manual process needs to be accounted for in the API.

## COMMON PITFALLS

As noted above, there are many reasons why an API fails to operate as intended. Sometimes the error can be traced back to initial discovery; in other cases, challenges occur during implementation. Common pitfalls include the following:

- States often focus the research and discovery phase on interviews with their own, internal staff. While understanding the state staff journey is essential, it is equally important to map the end-user journey. If you do not interview providers (or parents, if your API is designed to support the subsidy application process) you will miss important information about how these end-users interact with the system. Missing this perspective can lead to edge cases during testing, pilot, or after launch.
- Ensuring that required data points are available throughout the process is essential. If an API, or the provider or parent using the API, cannot see the required data (such as their children IDs) the process will fail at that gap.
- Vendor alignment is key and should be secured at the Request for Proposal or contracting stage to make sure your goals are the same as the vendor deliverables. Vendors can be helpful partners in successful implementation however API expectations, and funding aligned with the expectations, should be included in procurement and reinforced at every step in the process.
- Vendors must understand that they are expected to support all data fields required by the state. API launch will be delayed if additional fields or extra functionality are added at a later day and not included in the workplan or budget.

## TRACKING IMPACT: POTENTIAL METRICS TO CONSIDER

States will want to not only ensure that an API is effectively implemented, but that the change produces the desired outcome. To this end, it is important to think ahead about key metrics that can be gathered to deepen understanding and guide future revision. Some suggestions are included below.

- Track error rates, to determine if the API has reduced the number of errors in the system. States that have implemented APIs for child care subsidy billing report that error rates declined significantly when the API was used.
- Engage in a time study to measure the amount of staff time the API saves. This will require baseline data on the amount of time currently required to complete a task, compared with the amount of time required after implementation of an API.
- Cost savings can also be measured by converting time into money. When staff spend less time correcting errors, administrative costs decline. When data are entered only once, instead of multiple times in multiple systems, administrative costs decline.
- Gather information on satisfaction, via staff, provider and/or parent surveys. States that have implemented APIs for attendance tracking find that providers, parents and compliance staff are pleased with how much easier the process is after automation is successfully implemented.

Building data bridges is a very effective way to streamline administration of a host of routine tasks, and states engaged in this process have experienced significant wins. This [issue brief](#) highlights the Iowa experience with an API for subsidy billing. Opportunities Exchange is committed to helping states deepen engagement with modern technology with an eye to making the process easier and less time consuming for providers, families and other end-users. If your state is considering new technology to streamline subsidy administration, licensing, Quality Rating, or other child care supports, please visit the OppEx website pages focused on the [ECE Technology Ecosystem](#) and [Child Care Management Software](#) or reach out to any of our team members. 